PRODUCT-INFORMATION

2007

www.port.de
www.epl-tools.com

port

Software + Hardware

CAN+EPL

Professional tools

Product Information

II / *2007*

Development Environment and Software
for

CAN/CANopen/*DeviceNet/Ethernet*

Ethernet POWERLINK

http://www.port.de — http://www.epl-tools.com

port

Contact us:

eMail:	service@port.de
	newsletter@port.de
WWW	http://www.port.de
	http://www.canexperts.de
	http://www.epl-tools.com

port Gesellschaft für computergestützte Automation mbH
Regensburger Straße 7b
D-06132 Halle / Saale
Germany

Phone:	+49 - 345 - 77755 - 0
Fax:	+49 - 345 - 77755 - 20

Company Profile 1

Software Components CANopen 2

CANopen Tools 3

CANopen Starter Kit 4

Software Components DeviceNet 5

CAN Diagnostic Tools 6

CAN Interface Boards 7

Software for Ethernet POWERLINK 8

Hardware for Ethernet POWERLINK 9

Information 10

Table of Content

1. Company Profile

Company ..... page 7
Partners und Distributors ..... page 10
Services ..... page 11
Literature ..... page 12
Reference List ..... page 13

2. Software Components for CANopen

Introduction to CANopen ..... page 15
CANopen Library ..... page 17
CANopen Library Extensions ..... page 21
CANopen Library Driver Packages ..... page 24
CANopen Evaluation Software ..... page 28
CANopen Bootloader ..... page 30
CANopen Gateway Server CiA309-3 ..... page 32

3. CANopen Tools

CANopen Design Tool ..... page 33
CANopen Device Monitor ..... page 36
CANopen Configuration Manager ..... page 38

4. CANopen Starter Kit

CANopen Linux Starter Kit ..... page 41

5. Software Components for DeviceNet

DeviceNet Slave Development Kit ..... page 43

6. CAN Diagnostic Tools

CAN-Analyzer CAN-REport ..... page 45
CAN Bus Tester II ..... page 48
CANwatch ..... page 50

7. CAN-Interface-Boards

AT-CAN-MINI ..... page 51
EtherCAN Gateway ..... page 52
IGW/900 Gateway ..... page 54
USB/CAN-Interface CPC-USB ..... page 55
USB/CAN-Interface USB-BASIC+ XS Set ..... page 56
PCI/CAN-Interface CPC-PCI/PCIe ..... page 57
PCI/CAN-Interface PCI-IntelliCAN ..... page 59
can4linux - Linux Device Driver ..... page 60
can4vxworks - Device Driver for VxWorks ..... page 61

8. Software Components for Ethernet POWERLINK

Introduction to POWERLINK ..... page 62
POWERLINK Library ..... page 64
POWERLINK Starter Kit ..... page 67
POWERLINK Design Tool ..... page 69
POWERLINK Device Monitor ..... page 72
POWERLINK-REport ..... page 74
POWERLINK EDS-Editor ..... page 76

9. Hardware for Ethernet POWERLINK

POWERLINK Managing Node for PCI Bus ..... page 78
POWERLINK Gateway ..... page 79
POWERLINK MAC VHDL Code ..... page 80
POWERLINK Hub VHDL Code ..... page 81

10. Information

Directions to *port* ..... page 82

Company

Company Development

Foundation

port GmbH was founded in May 1990 as an independent company in Halle, Germany.

The aim in founding the business was to establish ourselves as a computer retailer for specific logic controller, track alignment and exerciser systems on industrial PCs. We offered solutions for decentralized, distributed systems within automated solutions which were based on special system concepts for the hardware and software development sectors.

Mission

Our mission is to develop the latest high-value technological products and services which are based on CAN, CANopen, DeviceNet und POWERLINK systems and can be used for industrial communication purposes.

Besides our standard high-performance products, we also develop customized, client-specific fieldbus-orientated solutions, which best meet clients demands.

Our services range from consulting (for fieldbus solutions) and on-site implementation of customer-specific embedded systems, to the processing of a wide range of customer projects for drives and distributed systems.

We see ourselves as an independent integrator of systems and use, wherever possible, open standards.

Experience

We now incorporate our long-gained experience within the business sector into the development of our products - and our customers end up benefiting as a direct result. Our software products have reached an advanced level of functionality, through having met the high technical quality standards of our world-wide clients.

Customer Orientation

The future of our business can only be formed together with our customers. We are constantly aware of this fact. We want, therefore, to be of maximum use to our customers by offering high quality products and competent advice.
We want to gain our customers trust by offering attractive products, excellent quality, innovative ideas, and develop a strong, healthy relationship with our customers.

Our aim is to create the conditions necessary to meet all customers needs and to win their trust, thus ensuring a continued, and above all, successful, co-operation which will last for many years to come.

We feel a close co-operation and continued exchange of information between customer and developer is of the upmost importance.

Software

Our software developers can draw upon several years of experience, and have acquired a wide range of expertise to deal with customer-specific tasks from having worked on complex projects. The customer projects, once realized, will always be in need of serial communication via fieldbus systems and data networks, no matter whether the task involves:

- development of motion and drives components
- development of driver packages
- configuration and diagnosis tools for CANopen- or POWERLINK- applications

Hardware Products

In order to offer customers not just software services, but also the required hardware products - which always meet current technological standards - our software products are compatible with the general interface which we share with a group of our associates.

Alongside our range of available software products, we offer a comprehensive assortment of CAN interface boards. Customers can then select the most appropriate variant.

Many Interface cards which come complete with their own processor or 'passive' cards that are operated by the main PC CPU, can be used for PCs with ISA or PCI bus. The widespread USB interface has also brought a lot of effect over the last few years. Besides USB adapters and Ethernet/CAN-gateways are supported.

CiA

port has been a member of CiA (CAN in Automation e.V.) for many years. As an active and competent business dialog partner we have a decisive influence on the formation of the latest developments especially of drives profiles. ***port*** chairs the special interest group CiA309 "Interfacing CANopen with TCP/IP".

Above all, as a result of its co-operation on the blueprint for the CiA309, port has a decisive influence over the formation of the process of linking up fieldbus technology with the ever-advancing internet-technology.

Our CANopen library is constantly brought up to date with the latest standards as we co-operate as part of the CiA working body, and also via the experience from various research and industrial projects that are supervised by our development engineers.

EPSG

Since 2004 we have been a member of the EPSG (Ethernet POWERLINK Standardization Group).

ETG

Since 2005 we have been a member of the ETG (EtherCAT Technology Group).

Through being a member of EPSG and ETG and from having developed new products in the internet technology sector, we are now able to provide customers with all-inclusive solutions which are automated by linking up to Ethernet.

Customers can also benefit from a wide range of technological solutions which can be used for project work.

Quality assurance

The aim of our product development and customer-specific services is to satisfy customers.

Happy Clients - in real-time communication and control

ZERTIFIKAT ◆ CERTIFICATE ◆ 認証証書 ◆ СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

TÜV SÜD Management Service

CERTIFICATE

The Certification Body
of TÜV Management Service GmbH
certifies that

port Gesellschaft für computergestützte Automation mbH
Regensburger Straße 7, D-06132 Halle

has established and applies
a Quality Management System for

Rapid Prototyping, Product Optimisation and Technology Transfer for Systems and Software Development

An audit was performed, Report No. 70008738
Proof has been furnished that the requirements according to

ISO 9001: 2000

are fulfilled. The certificate is valid until 2008-05-22
Certificate Registration No. 12 100 10910 TMS

Munich, 2005-05-31

TGA-ZM-07-92

TÜV Management Service GmbH · TÜV SÜD Gruppe · Zertifizierstelle · Ridlerstrasse 65 · 80339 München · Germany

Quality is assured with our process management system which is certified by ISO 9001:2000. In the field of customer support this system also offers a facility allowing customers an additional method to report problems not just via phone and email, but also via our website. After completing an online form, *(http://www.port.de/engl/reports/report_form.html)* this problem notification is forwarded directly to a competent member of staff and the end user then promptly receives the desired support.
Our version management system enables the restoration of software and documents in their respective versions.

General terms of business

Our terms and conditions are separated according to the specific product range:

- Terms and Conditions for the Delivery of Hardware and appendant Software
- Terms and Conditions for the Use and Maintenance of Software Products

The terms and conditions which apply to your product can be send on request by fax (+49 345 7775520) or Email (service@port.de).

Furthermore we provide the opportunity to download our general terms of sale as pdf-file from our web site.

Terms and Conditions for the Delivery of Hardware and appendant Software
http://www.port.de/engl/company/btc.html

Terms and Conditions for the Use and Maintenance of Software Products
http://www.port.de/engl/company/btc.html

Partners and Distributors

ECS Srl, Italy

ECS Srl is an Italian company in the development of embedded heating and cooling systems. It provides services for embedded controllers networked with CAN technologies.

http://www.ecsproject.com
mailto:service.CANopen@ecsproject.com

EMS Dr. Thomas Wünsche

Since the year 2000 ***port*** works together with the company EMS Dr. Thomas Wünsche. EMS delivers CAN products and software for a wide range of industrial applications.

http://www.ems-wuensche.com
mailto:marketing@ems-wuensche.com

Embedded Solutions AG, USA

Embedded Solutions supports and distributes a range of products in the USA. The ***port*** tools are supported on their ARM7 and ARM9 based products. ES AG runs Embedded Linux and Windows CE.

http://www.es-usa.com
mailto:info@es-usa.com

I+ME Actia

I+ME ACTIA GmbH is a 100% subsidary of the French ACTIA Group. I+ME ACTIA is a specialist in the area of micro electronics and serial networks (like CAN, VAN, J1850, ...).

http://www.ime-actia.de
mailto:info@ime-actia.de

Neomore, France

Neomore offers open, efficient and innovative solutions for fast development of embedded and real time applications.

http://www.neomore.com
mailto:sales@neomore.com

Pyramid Solutions Inc., USA

port is using the DeviceNet experiences of Pyramid Solutions to provide a complete spectrum of CAN protocols in Europe. Pyramid Solutions is a recognized leader in software engineering and systems integration, with a comprehensive technical staff.

http://www.pyramidsolutions.com
mailto:gbeal@pyramidsolutions.com

SI-Kwadraat, Netherlands

The company SI-Kwadraat is a qualified partner for all questions regarding CAN technologies in the Benelux countries.

http://www.si-kwadraat.nl
mailto:info@si-kwadraat.nl

Sistemas Embebidos SA, Spain

For all Spanish speaking countries SISTEMAS EMBEBIDOS S.A. - España - is your partner. Sistemas Embebidos is a well-known Systems Integrator in Spain.

http://www.embebidos.com
mailto:sistemas@embebidos.com

Warwick Controls WCT, UK

Founded in 1999, Warwick Control Technologies have been providing unparalleled expertise, tools and development services for embedded controllers and systems within the automotive and industrial automation industries.

http://www.warwickcontrol.com/
mailto:enquiry@warwickcontrol.com

Services

port has a long experience in usage of CAN/CANopen and POWERLINK fieldbus technology for embedded systems. We provide products like the *CANopen Library*, the POWERLINK *Library* as well as helpful tools for development of CANopen, POWERLINK devices and working with applications that make use of CAN/CANopen or POWERLINK.

We do not only offer our products to customers but offer our customers support for handling, application engineering and implementation of CAN/CANopen and POWERLINK projects in form of engineering services, trainings and initial operation.
Support in this form is very efficient and reasonable when the project schedule is tight.

Software development

Services

port provides the following engineering services:

- implementation of the CANopen and POWERLINK communication profile
- implementation of CANopen device profiles
- implementation of DeviceNet device profiles
- consultation in the field of CAN, CANopen and POWERLINK
- application-specific implementation and extensions
- customer-specific Embedded LINUX™ developments for Ethernet

The success of ***port*** is based on solutions for tasks of different areas of controlling systems that cannot be solved with standard software.

Benefits for our customers

We are of benefit for our customers by:

- long standing experience in the industrial sector of fieldbus technology
- good expertise as provider of software based solutions
- high flexibility in development of customer specific solutions
- flexible adoption of products of ***port*** to customer requirement
- on-time realization of challenging customer projects

Starting with the definition of the requirements and continuing with steps for system design, implementation and integration up to software testing ***port*** provides all its customers a complete development from one source. At all steps contact to the customer plays an important role. This guarantees adherence to the project schedule and adherence to the planned costs as well as security in reproducibility of the already gained development steps.

Training

CANopen and POWERLINK

port offers customer specific training concerning CANopen and POWERLINK. With this training we provide an efficient introduction into the topics of CAN/CANopen and POWERLINK for participants. We can adjust the contents of each training to the customer needs and emphasize on fixed topics. The aim of our training courses is to provide a good knowledge base to our customers for a good start off and at carrying out CANopen\POWERLINK projects with the *CANopen Library*, POWERLINK *Library* and all available software tools.

The topics of the training courses are:

- Introduction to CAN and CANopen or POWERLINK (CAN PHY, DLL, CANopen services and profiles)
- Description of the protocol library
- Description of the driver interface
- Description of the CANopen/POWERLINK Software Tools
  (Analyzer, Configuration, Design Tool)

Please request our detailed training program.

DeviceNet

port offers training courses for the DeviceNet protocol on request.

Integration Support

For the implementation of CANopen/POWERLINK into the customer-specific application or the adaption into a special hardware we offer implementation support for 3-days with the following topics:

- Introduction into CAN/CANopen or POWERLINK (see Training)
- Introduction into the library with an example for implementation and usage of the Software Tools of ***port***
- transfer to customer-specific hardware

To realize the CANopen or POWERLINK implementation in such a short time the co-operation with our customers is indispensable. Support by the customer is done by:

- the functioning hardware
- the co-operation of a software engineer from the client with special knowledge about the used hardware

For the CANopen or POWERLINK implementation of a specific customer hardware it may be necessary that the customer needs to provide us with a suitable development environment (compile, debugger etc.).

Literature

The studies of books available on a specific topic is a kind of scientific communication from the experienced user to the novice.

With it everybody is enabled to gain from available knowledge and existent experiences.

This means that existent knowledge can be transformed into individual knowledge and skills. The study of literature therefore is the fastest possibility to gain from others experience.

The following paragraph presents an overview of available sources of literature.

Articles

Design and Implementation of CANopen devices
Rüdiger Härtel
9th international CAN Conference Munich 2003

The CiA CANopen Gateway Protocol
Heinz-Jürgen Oertel
CAN Newsletter, CiA, March 2004

Safety in CANopen
Rüdiger Härtel (***port*** GmbH)
SPS Magazin HMI-Special 2005

Device description in XML-format
Thilo Shumann (CiA), Torsten Gedenk (***port*** GmbH)
etz issue 8/2005

How to implement a CANopen drive according to CiA402
Torsten Gedenk
CAN Newsletter Special Edition "Motion Control"
November 2005

Using modern CANopen development tools effectively
Heinz-Jürgen Oertel
internal publication, ***port*** GmbH, 2004

Where is CANopen heading? The CANopen protocol on the test bench.
Heinz-Jürgen Oertel
MSR Magazin issue May 2005

Rapid prototyping for CANopen system development
Rüdiger Härtel
11th international CAN Conference Stockholm 2006

Develop CANopen devices the easy but safe way
Heinz-Jürgen Oertel
CAN Newsletter March 2006

How much Power for Ethernet Powerlink?
Heinz-Jürgen Oertel
Elektronik issue October 2006

Controlling CANopen nodes remotely via Ethernet
Torsten Gedenk
CAN Newsletter March 2007

Books

"CANopen" Library User Manual
paperback, app. 225 pages, English

The User Manual serves as an introduction for using the source code library. The procedure to use this *CANopen Library* and the process of integration into the customer's application are described here. This document describes the properties of all distributions of the *CANopen Library*.
ISBN 3-8334-4621-8

CANopen
Holger Zeltwanger
publisher VDE Verlag GmbH,
Berlin/Offenbach, 2001
with CD-ROM, ISBN 3-8007-2448-0

Controller Area Network
Basics, Protocols, Modules, Applications
Prof. Dr.-Ing. K. Etschberger
Hanser Verlag, 2000, second edition, hard cover, CD
ISBN 3-446-19431-2

CAN: Controller Area Network
Basics and practice.
Prof. Dr. Wolfhard Lawrenz
hard cover edition - 457 pages (2000)
Hüthig, Heidelberg; ISBN: 3778527800

CAN System Engineering
From Theory to Practical Applications
Prof. Dr. Wolfhard Lawrenz
hard cover edition - in English - 520 pages
Springer Verlag ISBN: 0387949399

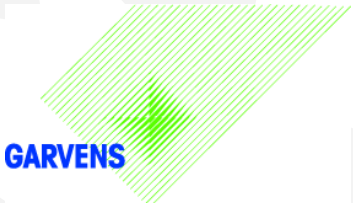
WorldWideWeb

http://www.CAN-CiA.org/
http://www.CAN-wiki.info/
http://www.ethernet-powerlink.org
http://wiki.tcl.tk/
Please visit our large link collection at:
http://www.port.de/engl/publicat/can_links.html.

Reference List

On this page we would like to present a list of our (main) customers of our products and references for projects done for customers. Please visit our larger list on the ***port*** web site http://www.port.de/gif/references/kreferences.html.

Leica Projektion GmbH
Zett Geräte



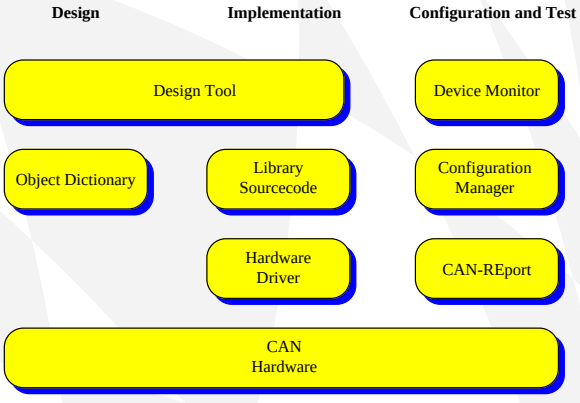
WITTENSTEIN AG

And a great many more .....
Happy Clients in Realtime Communication and Control on five continents.

Introduction to CANopen

CANopen is a CAN-based application layer protocol according to the ISO/OSI model. It was developed as a standardized embedded network with highly flexible configuration capabilities. CANopen was designed for motion-oriented machine control networks, such as handling systems. By now it is used in many various fields such as medical equipment, off-road vehicles, maritime electronics, public transportation, building automation, etc.
The essential communication services are defined in the CANopen communication-profile (CiA 301). Furthermore there are several device, interface and application-profiles (CiA 3xx, 4xx), that are administered by CAN in Automation e.V. (CiA - international users' and manufacturers' group). The current version 4.xx of the CANopen communication-profile is standardized as EN 50325-4.

Since 1995 ***port*** has committed itself to active development and maintenance of standards within CiA and among other things as chairman in the interest group *IG Drives and Motion* and *IG Gateways*. With these experiences CANopen products of ***port*** are constantly enhanced, improved and tailored to customer requirements.



The tools are coordinated and can be employed in all phases of device development - from design, test to configuration of complete networks.

CANopen Sourcecode Library

The *CANopen Library* provides all required services for a CANopen compliant communication and facilitates easy and fast development of devices for the different supported hardware platforms.

Through its modular structure the function range can be extended to the optimum with additional modules (LSS, redundancy, flying manager, CANopen safety, ...) to match the customer requirements. Thus, slave devices as well as master devices can be implemented smoothly as single and multi-line devices.

Delivery comprises the configuration software *CANopen Design Tool Light*. With it the required services are scalable to a large extend and available application resources are treated with care.

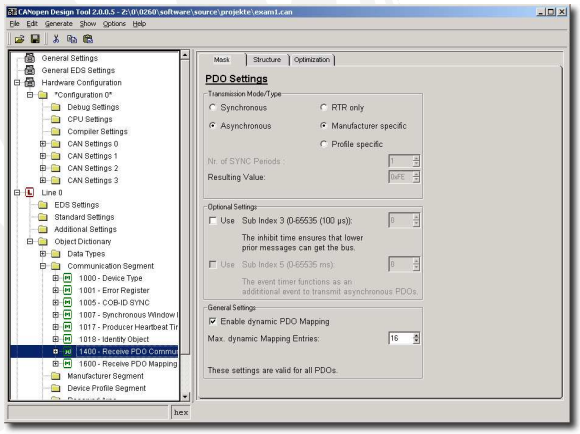
The strict ANSI-C conformity and the separation of the application layer from the data link layer eases porting it to arbitrary hardware platforms allowing for retention of its high performance. The result are portations to 8/16/32 bit controllers, little- or big-endian controllers, micros or DSPs.

A comprehensive documentation including user manual and reference manual as well as several ready to run examples also belong to the delivery of the *CANopen Library*.

CANopen Design Tool

The *CANopen Design Tool* (CDT) is used for administration of CANopen projects. With its graphical user interface it simplifies

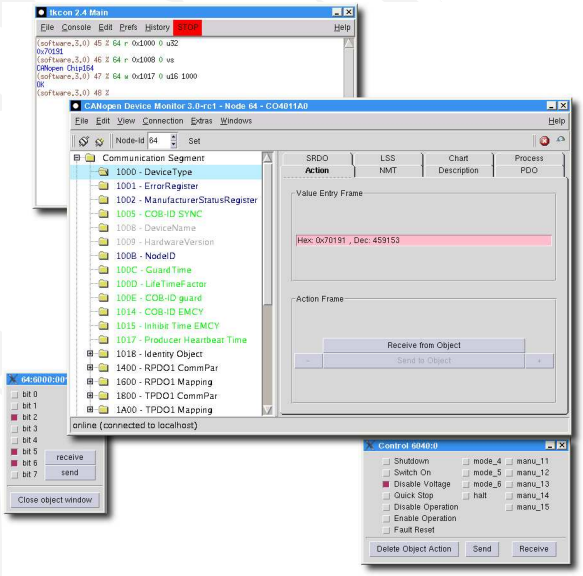
- creation of the object directory,
- creation of EDS files (standard or XML format)
- configuration of *CANopen Library* services and setting of hardware properties.



Header files are generated that are included in the *CANopen Library*. The advantages of this tool do not only lie in consistency of device implementation and documentation, but also in easy reusability and avoidance of redundant and error prone work, which in turn leads to an accelerated development.

CANopen Device Monitor

The *CANopen Device Monitor* (CDM) is a tool for graphical inspection and configuration of CANopen devices. With the information of EDS files all objects of devices can be accessed. Every modification of variables of devices can be evaluated and be stored as DCF file.



In addition to the NMT commands like OPERATIONAL, PRE-OPERATIONAL, STOPPED, .. there are network control mechanisms like heartbeat monitoring or node guarding as well as special functions like SYNC, TIME, EMCY, .. available.

An integrated scripting language provides access to all implemented CANopen services. Thus test applications for the device development and control applications can be developed with minimum effort.

The graphical user interface and the CANopen server are separate applications and communicate over TCP/IP network connections. For this reason it is not necessary any longer that both programs reside on the same computer. Through the communication over TCP/IP according CiA 309-3 both components can be connected over Internet or dialup connections. This allows easy and cheap remote maintenance.

The CANopen server contains a complete NMT master functionality and provides simultaneous access from multiple TCP/IP clients.

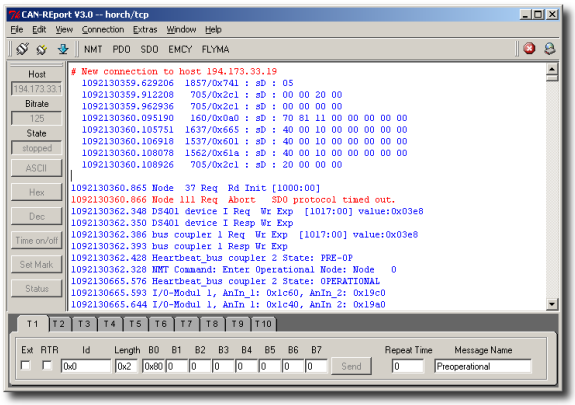
Configuration Manager

The *CANopen Configuration Manager* (CCM) has been developed for configuration tasks of complete CANopen networks. Besides the PDO linking, i.e. defining the PDO communication relations, node control settings can be carried out.
The communication relations can be defined between arbitrary sender and receiver nodes, if appropriate entries are available in the object directory.
The required information is obtained from EDS files of the network nodes. DCF files can be generated for all configured communication relations of the EDS data. Furthermore the configuration can be stored as a C data structure that can be included easily into a master application.

For access on the CAN bus the *CANopen Configuration Manager* makes also use of the CANopen Server and therefore allows remote access too.

CAN-RE*port*

The CAN-RE***port*** is an efficient and versatile usable CAN-Analyzer. It comes with powerful functions for monitoring the CAN bus traffic with different trigger possibilities, storage of CAN messages into log files and sending of CAN messages.



For the CANopen protocol, messages can be displayed service oriented in separate logging windows. Furthermore it is possible to develop application extensions for interpretation of proprietary protocols. The integrated scripting language allows universal usage during development, test and maintenance.

The CAN-RE***port*** consists of the CAN hardware interface and the visualization software. Both are connected as server and client by a standard TCP/IP network connection. This separation makes it possible to use the CAN interface as a remote interface, i.e. both parts can be located at different computer systems. Remote monitoring of CAN networks is possible without additional or modified software over LAN, dialup or Internet connections.

Overview

The *CANopen Library* provides all required services for a CANopen compliant communication according to the communication profile CiA 301 V 4.1. It facilitates easy and fast development of master and slave devices and is able to serve one or multiple CAN-Controller in one device.
The *CANopen Library* is available in different expansion stages:

- Slave Small
- Slave
- Master/Slave

The functional range can be extended by additional modules (LSS, Redundancy, Flying Master, CANopen safety, ...) (see *CANopen-Library Extension modules*).

Access to the hardware is carried out via a defined driver interface, which is available for many CPU- and CAN controller with and without operating system (see *CANopen Driver Package*).

Application

The *CANopen Library* is based on ground of the communication profile CiA 301 V4.1 of the CiA e.V. "CANopen Application Layer and Communication Profile" and EN50325-4 , respectively, and provides all services specified therein. It is completely written in ANSI-C and can be compiled with every ANSI-C compliant compiler.

Depending on the required scope of functionality the *CANopen Library* is available in different expansion stages. For development of small sensors and actuators with limited CANopen services the *Slave Small*-Version is available. Its limitations are composed of a restricted number of service instances and no support for the CANopen services SYNC and TIME.
With the *Slave*-Version of the *CANopen Library* all services are provided for development of full-featured slave devices.
The functionality of the network management master as well as the comfortable node monitoring functionality is provided by the *Master/Slave*-Version, which of course contains the services of the *Slave*-Version.

Additional services of other communication profiles (CiA 3xx) are provided by means of extension modules.

All versions of the *CANopen Library* are compatible to each other and are constantly tested with the current CANopen Conformance Test for compliance with the standard.

	Slave Small	Slave	Master/ Slave	Extra Package
SDO Server	32	128	128	
SDO Client		128	128	
SDO Segmented Transfer		✓	✓	
SDO Block Transfer				✓
Dyn. SDO Slave				✓
SDO Manager				✓
Program Download		✓	✓	
PDO Consumer	64	512	512	
PDO Producer	64	512	512	
Dynamic Mapping		✓	✓	
Bitwise Mapping		✓	✓	
MPDO Source Mode				✓
MPDO Dest. Mode				✓
Nodeguarding Master			✓	
Nodeguarding Slave		✓	✓	
Lifeguarding		✓	✓	
Heartbeat Consumer		128	128	
Heartbeat Producer	✓	✓	✓	
EMCY Consumer		128	128	
EMCY Producer	✓	✓	✓	
Time Consumer		✓	✓	
Time Producer			✓	
SYNC Consumer		✓	✓	
SYNC Producer			✓	
NMT Slave	✓	✓	✓	
NMT Master			✓	
NMT Flying Master				✓
Bootup Procedure			✓	
Configuration Manager			✓	
Safety Communication				✓
Redundancy Support				✓
LSS Services				✓
LED Indicator	✓	✓	✓	
CiA 401 Framework				✓
CiA 402 Framework				✓

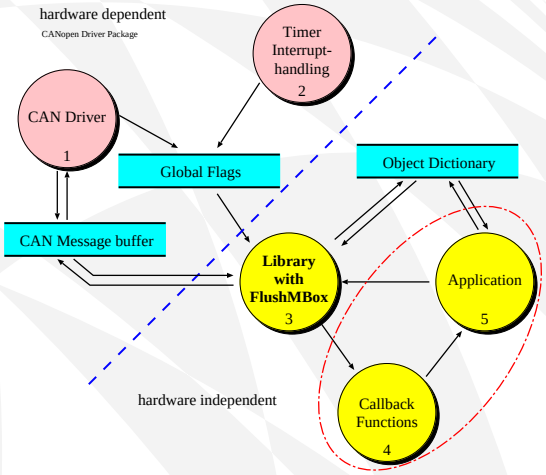
All hardware specific parts are decoupled from the CANopen protocol stack through a defined driver interface. This provides easy adaptation to different hardware platforms.

The user application communicates with the *CANopen Library* through function calls and call-back functions.

Configuration and scaling of the *CANopen Library* is done with the help of the *CANopen Design Tool*,

which is delivered as *CANopen Design Tool Light* version. With it the *CANopen Library* can be tailored to an optimum to the available resources of the application. Besides the creation of the object directory all settings for the hardware can be carried out with it.

Description



The *CANopen Library* consists of a hardware independent and a hardware dependent part that communicate with message queues. The hardware dependent part consists of controlling software for the CAN controller and timer functions. A detailed description can be found in the *CANopen Driver Package*.

The application communicates only with the hardware independent part of the *CANopen Library*. That way it is possible to exchange drivers without any influence on the functionality of the application. The initialization of CANopen services is done with function calls within the application. During the execution of the application the *CANopen Library* executes all necessary communication tasks autonomously and informs the application about CANopen messages with the help of callback functions.
Communication requests from other devices as well as necessary periodical tasks and time out monitoring is handled within the *CANopen Library*. All requests are proved for correctness (access rights, data types etc.). The application is notified after completion of the communication and occurrence of failures, respectively, through service oriented callback functions. In these callback functions appropriate actions can be carried out from the application.

The object directory is designed to contain references to the variables in the user application. Consequently it is possible to take over variables from an existing software without any changes in the object dictionary.

The high degree of scalability of the *CANopen Library* is of particular importance for devices with limited resources. On one hand, this is achieved by the modularity in individual service groups, like *sdo.c*, *pdo.c*, *...sync.c*, and on the other hand, through the use of compiler directives in the respective modules.

Thus, the code size is proportional to the utilized CANopen services.

Furthermore, variants of the *CANopen Library* for the support of multiple CAN lines (max.255) are available. Therefore it is possible to serve several independent CAN networks on devices without an operating system or with an operating system. The usage of an operating system requires it to provide means of resource protection mechanisms. Each line holds its own object directory and can be used as master and slave, respectively, independent of the other lines. Owing to the separation of the protocol stack and the hardware drivers, the individual lines can also be operated with different CAN controllers.

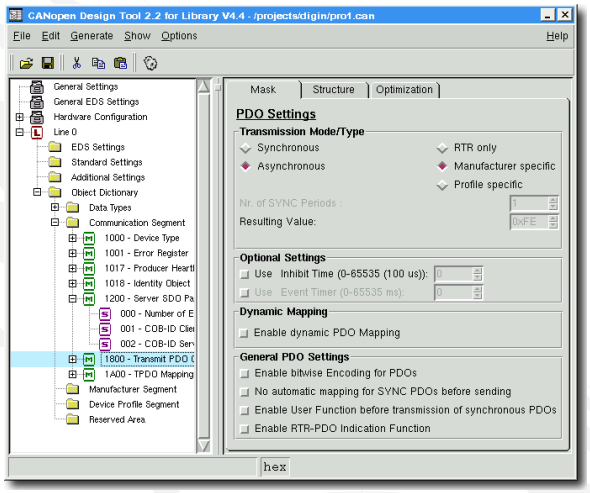
Delivery of the *CANopen Library* comprises different example programs that describe the usage of the various CANopen services. All examples contain a complete implementation of a CANopen device including the object directory as well as application code. These are ready to be compiled and run.

Among the detailed documented source code there is the reference manual and an a printed user manual, numbering 200 pages, as documentation of the *CANopen Library* available.

The *CANopen Library* is constantly improved and adopted to customer requirements. In order to keep up with the latest version of the *CANopen Library* ***port*** provides its customers an update service.

The support engineers of ***port*** are ready to answer all questions by email, phone or fax regarding the initial operation of the *CANopen Library* or further questions to the topic CANopen. This service is exempt from charges up to 6 months for requests by email and fax and up to 6 months for requests by phone.

For development, test and initial operation of CANopen devices ***port*** provides a comprehensive tool chain. The creation of the object directory is simplified with the *CANopen Design Tool* (*CANopen Design Tool Light* version belongs to the scope of delivery).



The graphical *CANopen Design Tool* is available for the creation of the object directory, the electronic data sheet (EDS file) and for the documentation of the device in HTML or text format.

The *CANopen Device Monitor* can be utilized for commissioning, but also for the implementation and test phase. As a fully implemented CANopen master the *CANopen Device Monitor* with a graphical user interface assures access to all services in the network and also allows analysis of the bus traffic.

For the development of CAN application monitoring and analyzing the CAN bus traffic is essential. The service oriented display of CANopen messages of the CAN-RE***port*** allows easy and quick interpretation of CANopen messages.

Besides the communication objects of the CiA 301 several application objects in different device profiles are also specified in CANopen. These definitions guarantee a defined behavior of the corresponding device class and enable the interchangeability of CANopen devices.
For these different device profiles ***port*** provides extension modules as an add-on to the *CANopen Library*. These modules allow the customization of an application based on the CANopen standards.
All modules are available in ANSI-C source code and can be used with all versions of the *CANopen Library*.

Please have a look at the information of the data sheet *CANopen-Library Extension modules*.

System environment

The *CANopen Library* runs on targets with and without operating systems. It supports many of the available CAN controllers and many microcontrollers/processors. For detailed information see data sheet *CANopen Driver Package*.

For systems based on Windows™ the *CANopen Library* is also available a Dynamic Link Library (DLL). This DLL is available for special hardware boards and contains all services of the *Master/Slave standard* edition of the library. It can be used to create master or slave applications.

Furthermore CANopen-Starter Kits and a free CANopen Evaluation software on CD are available on request.

Delivery scope

- CANopen library with separate driver interface
- CPU/CAN driver
- numerous, immediately compilable examples
- *CANopen Design Tool Light*
- detailed user manual
- reference manual containing descriptions of all functions, including parameters and return values
- 6 months support by telephone free of charge
- 6 months support by E-Mail free of charge
- 6 months update service free of charge

Licensing conditions (excerpt)

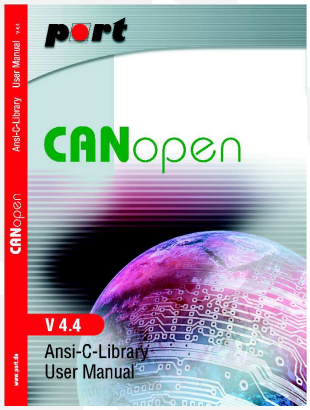
For the *CANopen Library* a one-off license fee is charged in form of the purchase price. Further license fees do not arise from the deployment of the software within the same company (no runtime licenses).
It is not allowed to hand over the software and the implementation, respectively, towards a third party.

Further Services

- Maintenance Agreement
  All changes of current standards as well as extensions through new developed standards are constantly incorporated into the *CANopen Library*. In order to take profit of the changes ***port*** offers all its customers a maintenance agreement with the following conditions:
  - ☐ updates free of charge for the contracted period
  - ☐ free of charge support for the *CANopen Library* and topics of CANopen
- Support for initial operation
  In order to provide a quick and effective access to the development of CANopen devices we recommend to do the initial operation of the CANopen device together on the target platform. Customer experiences of his/her hardware and the used compiler and the experience of our engineers with

CANopen and the *CANopen Library* can complement each other. This leads to reduced development times and a CANopen conform device.

- Conformance Test
  In preparation of the charged Conformance test of the CiA we provide you our service, knowledge and equipment. With the preparation and execution of preliminary tests possible problems can be detected and removed.
- System analysis, consulting and support
  Take profit of our Know How during planning and development of your CANopen devices and networks. An ideal design requires knowledge about the used protocols and the system environment. Our competent engineers create together with you a cost-efficient solution that fits your needs to an optimum.
- Software development services for CAN, CANopen and DeviceNet
- Training
  To simplify the first steps with a new technology ***port*** provides one-day and several-days trainings for the topics of CANopen and its services as well as for the *CANopen Library* and its tools. These trainings can be given at ***port*** or at your companies rooms and are exclusively carried out for your workers.
- CANopen User Manual
  You want to get to know about the advantages of our *CANopen Library*? Then our user manual is the best starting point to get an overview about the functionality and deployment possibilities of the *CANopen Library*.



- Starter Kit

For a quick and uncomplicated start with CANopen the *CANopen Starter Kit* provides everything you need. It comes with a pre-compiled *CANopen Library* for your target hardware and several simple examples that provide a first insight and that ease the start with CANopen. As hardware interfaces several possibilities are avaliable, like PC cards, parallel port dongles, PCMCIA cards and USB interfaces.

With these setting up a CAN network is as easy as a children's game.

Further informations can be found at *CANopen LINUX™ Starter Kit*.

Ordering Information

0564/01	CANopen-SRCLIB Slave Small Single Line
0564/12	CANopen-SRCLIB Slave Single Line
0564/10	CANopen-SRCLIB Master/Slave Single Line
0564/15	CANopen-SRCLIB Slave Multi Line
0564/13	CANopen-SRCLIB Master/Slave Multi Line
0910/xx	CANopen-DLL Master/Slave Single Line
0920/xx	CANopen-DLL Master/Slave Multi Line
0564/90	CANopen-SRCLIB User Manual (Paperback)
0571/10	CANopen-SRCLIB Software Maintenance Agreement
0564/39	CANopen Evaluation-Software
0569/10	CANopen Training Course
0570/10	CANopen Integration Support
0572/01	CANopen Starter Kit for LINUX

CANopen Library Extensions

Overview

Besides the standardization of communication objects CANopen specifies application objects used in various device profiles. These device profiles guarantee a defined device behavior and provide thereby interchangeability of CANopen devices.

For employment of these profiles ***port*** provides extension modules for its *CANopen Library*. These modules make it possible to use the desired device profiles easily.

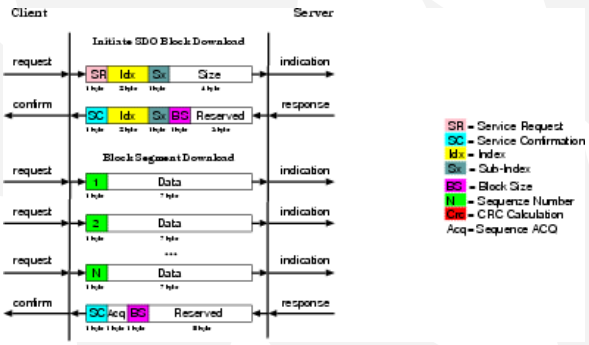
Currently ***port*** supports the following device profiles:

• CiA 301	SDO Block Transfer
• CiA 301A	Multiplexed PDO's
• CiA 302	Flying Master
• CiA 302	CANopen Redundancy Support
• CiA 302	SDO Manager/ SDO Requesting Devices
• CiA 303-3	LED Modul CANopen Indicator
• CiA 304	Safety-Relevant Communication
• CiA 305	LSS Layer Setting Services
• CiA 401	Generic I/O Modules
• CiA 402	Drives Support

All modules are available in ANSI-C source code and can be used with all versions of the *CANopen Library*.

Description

CiA 301 SDO Block Transfer



SDO transfers are based on the client-server model with a handshake after each transfer. For a larger block of data this will take a large amount of time. Therefore a new SDO mode has been defined in CiA 301 V4.x. It is called SDO block transfer. Using the block transfer a sequence of blocks can be transmitted without a large overhead. Each block is a sequence of up to 127 segments (e.g. CAN telegrams) containing only a sequence number and the data.

CiA 301A Multiplexed PDO's

If the application has a lot of data with the same properties a special PDO type can be used. It is called a Multiplexed PDO (MPDO). MPDOs transmit with every telegram the index and the sub-index of the given data. Therefore the maximum data length can be only 4 bytes. The transmitted index and sub-index can be the index and the sub-index of the producers object dictionary (MPDO Source Addressing Mode) or the index and the sub-index of the consumers object dictionary (MPDO Destination Addressing Mode).

CiA 302 Flying Master

The package *CiA 302 Flying Master* is an extension to the *CANopen Library* library of ***port*** and provides functions for implementing the flying master functionality for CANopen devices. Flying master means that a second device in a network can take over the master role when the master dropped out.

The provided functions fulfill all demands of the standard:

- determination of master device in a network
- detection of the active master
- priority driven master negotiation
- check for multiple master

The application is informed about changes of the master state with indication functions.

CiA 302 Redundancy Communication

For employment of communication systems in maritime applications a redundant bus system is required. Because only the communication has to be redundant all services provided by CANopen can be used by this package. With function calls the redundant

communication is handled automatically by the library. Information about other events in the different bus systems like drop out of one bus line, restoration of a bus line, missing communication and others are provided to the application with indication functions. Redundancy needs at least two CAN controllers. ***port*** offers implementations for stand alone CAN controllers as well as with microcontrollers with more than one CAN integrated.

CiA 302 SDO Manager/SDO Requesting Devices

SDO connections exist exclusively between Client and Server. For the operation of configuration and analysis software as well as HMI which are only temporarily present in a network static SDO connections are not possible. For these cases the use of dynamic SDO connections is foreseen. With the extension module *CiA 302 SDO Manager/SDO Requesting Devices* the *CANopen Library* of ***port*** can be expanded to use this functionality.

The module contains functions for implementing either an SDO manager or an SDO requesting device. All functions described in the standard are available:

- manage dynamic SDO connections (request, release, configure, monitor)
- manage the COB IDs of SDO
- manage the SDO connection table

The application is informed about events by means of indication functions.

Furthermore functions to request or release SDO connections are enclosed:

- register as an SDO requesting device
- request/release SDO connections
- request all default connections

The package contains source code, documentation and example programs for getting into it quickly.

CiA 303-3 LED Module

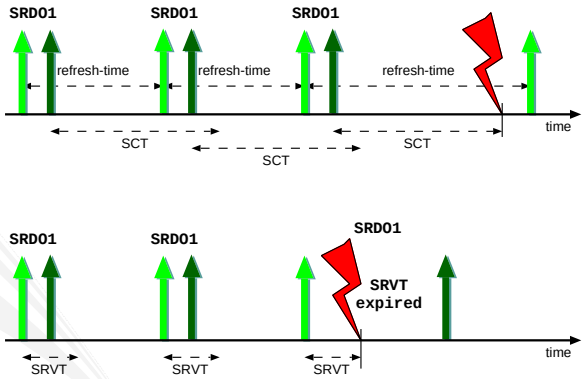
For a consistent state or error indication respective error diagnosis of CANopen devices the module *LED* is available. It is included in the default delivery of the CANopen Library.

According to the specification of CiA 303-3 a NMT-LED and an Error-LED can be used as two single colored LED or as one bicolored LED. The state machine for controlling the LED is realized within the module. The desired configuration can set with the configuration tool.

On state change of either NMT or error state the application is informed by an indication function which will then control the LED, i.e. switch on or off the appropriate LED.

The LED module is alltimes delivered by the library.

CiA 304 Safety-Relevant Communication



In order to transmit safety relevant data in a CAN network further measures have to be taken. This is done with this extension package. Communication takes place with so called SRDO i.e. Safety Relevant Data Object. These objects have communication and mapping parameter like PDO. The further measures consist of:

- sending data as plain and inverted data with different CAN identifier
- monitoring of sending data cyclically and in correct order
- protecting communication and mapping parameter with a checksum
- additional activation flag in the object directory

All monitoring functions can be realized with functions of the *CANopen Library*, but considering safety aspects monitoring functions should be implemented by the application developer.

Consistency of the data in the object directory is achieved through further safety measures.

- modification of communication and mapping parameter is only allowed in the state PRE-OPERATIONAL
- on modification all data lose their validity
- mapping data is also stored inverted
- a checksum calculated over all data of an SRDO is checked when changing into state OPERATIONAL

With this all planned safety measures of the standard are realizable.

CiA 305 Layer Setting Services

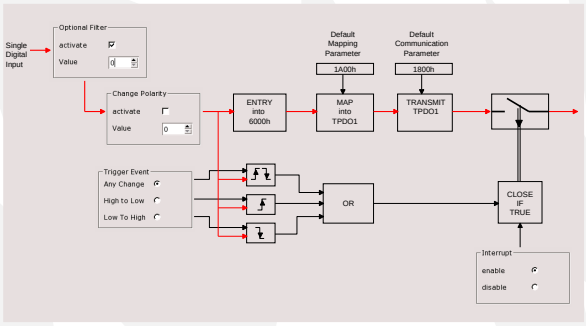
The extension package *LSS* (Layer Setting Services) for the *CANopen Library* contains all functions to operate CANopen devices as LSS Master or LSS Slaves. With it unconfigured devices in a network can be identified by their unique manufacturer, product, serial and revision number and configured. After identification bit rate and node id can be configured for each device.

Configuration with LSS can only be implemented on a CANopen master. With function calls existing slaves can be identified or addressed. In indication functions the responses of slave devices can be processed by the master.

Responses of the LSS slaves to LSS requests of the master are generated automatically by the *CANopen Library*. Only when new parameters like bit rate or node id are set the application will be informed with indication functions.

Delivery comprises source code, documentation and example code to get started quickly.

CiA 401 generic I/O modules



The device profile for Generic I/O-Module is an extension to the *CANopen Library* of ***port*** and provides functions for realizing I/O devices that conform to the device profile CiA 401. The implemented functions realize the functionality as specified in the standard, i.e. logic operations, polarity switching, filter mechanisms, limit monitoring and others. The developer has to provide functions to access the hardware, ie. reading inputs, writing outputs.
Delivery comprises source code and documentation.

CiA 402 Drives Support

Drives Support is available from ***port*** in different ways. First of all using the *CANopen Design Tool* and its CiA 402 profile data base. Second, from a lot of implementations we did, we have some give-away support functions in C combined in a CiA 402 framework. And third, our experience. Implementing drives and the CiA 402 profile is one of the most complicated programming tasks.

For the test of an drives implementation according to CiA 402 the *CANopen Device Monitor* with its drive specific plug-in is suggested. Look at the drive state machine or the different movement modes in a graphical way.

Ordering Information

0564/50	CiA 401 Source Code Generic I/O Modules
0564/51	CiA 401 Source Code and Profile Database for Generic I/O Modules
0564/52	CiA 302 Source Code Flying Master
0564/54	CiA 305 source code Layer Setting Services
0564/55	CiA 304 Source Code Safety-Relevant Communication
0564/57	CiA 307 Source Code CANopen Redundancy
0564/58	CiA 304 Source Code and Profile Database for Safety-Relevant Communication
0564/60	CiA 302 Source Code dyn. SDO Manager/ SDO Requesting Devices
0564/62	CiA 301A Source Code Multiplexed PDO's
0564/63	CiA 301 Source Code SDO Block Transfer

CANopen Library Driver Packages

Overview

The *CANopen Driver Packages* (DP) provide access to the hardware of the *CANopen Library* of ***port***. The Driver Packages are the keys for the usage of the target system independent of CANopen Library. They are designed for target systems with and without operation systems.

Description

The *CANopen Driver Packages* provide all necessary functions for

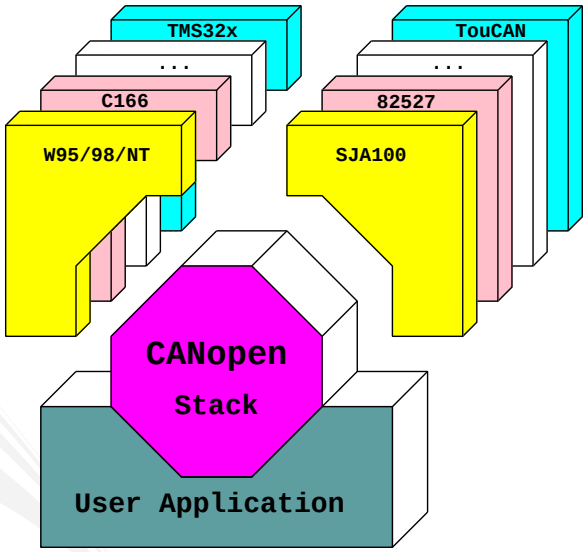
- initialization of the hardware
- controlling the CAN Controller
- message filtering
- message buffering and
- provision of a timer period

by means of a well defined interface. A *CANopen Driver Package* is comprised from a CPU- *(DP/CPU)*, a CAN- *(DP/CAN)* and where applicable a driver for an operating system *(DP/OS)*. These drivers can be used in any combination.

The following table shows hardware combinations of CPU and CAN drivers that are often used by our customers.

CAN Controller	microcontroller family																		
	Microcontroller independent ANSI-C driver	8051 Family	Infineon C5xxC	Dallas 80C390	Atmel T89c51	Infineon 16Cy	Intel 80x86	Motorola 68k	M 683xx	PowerPC	Fujitsu 16LX	68HCxx	TMS320Cxx	Mitsubishi	Motorola 56F800	ARM 7/9	M HC08/HC912
Integrated CAN		●	●	●	●	●			●	●	●	●	●	●	●		●
SJA1000/82c200	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	
Intel 82527	●	●	●	●	●	●	●	●	●	●	●	●	●	●			
TouCAN	●									●							
CANary	●	●			●												
msCAN	●											●			●		●
Bosch C_CAN		●														●	
Infineon TwinCAN						●											

All drivers are available as documented ANSI-C source code. This allows easy and quick adoption to system specific hardware and trouble-free compilation with all ANSI-C compliant ANSI-C compilers.



The comprehensive documentation ensures that the driver can be easily ported to another hardware platform by the user.

Configuration possibilities

The driver provides access to the CAN controller by several means:

- Memory Mapped Mode
  (addressing the CAN controller via the address space of the CPU)
- I/O Mapped Mode
  (addressing the CAN controller via the I/O address space of the CPU)
- Latched Mode
  (addressing the CAN controller via address latches)

The used access method for accessing the CAN controller register depends heavily on the used hardware and can be adopted by means of access macros.

The driver for the FullCAN controller types provide the usage of

- FullCAN-Mode
  filtering CAN messages in hardware for all channels available (1 channel for sending or receiving)
- FullCAN mode with 1 transmit channel
  one transmit channel for all transmit messages
  all other channels are configured as receiving channels using the hardware filter
- BasicCAN mode
  1 transmit channel for all CAN messages
  1 receiving channel for all CAN messages

The last two modes provide the possibility to use devices with FullCAN controller, that support more CAN objects as are provided from the hardware of the CAN controller.

Many drivers do contain code for

- step by step initial operation of the driver
- quick error detection
- extensions that have been added due to customer requests

that can be enabled via appropriate compiler defines.

Besides the drivers for processors with multiple CAN controller there are several other drivers that support configuration for multi-line usage with the *CANopen Library*. Thus there is a wide range of drivers that can be used for multi-line devices.

Please note that there are possible hardware constraints especially with older CAN controller.

Configuration of the driver is done with the *CANopen Design Tool*, which is enclosed to the *CANopen Library* as light version.

Resources

All drivers normally transmit and receive interrupt driven and therefore require the integration into the interrupt system of the device. It also can be used in polling mode for special-purpose applications.
The time base for the *CANopen Library* can either be provided by the *CANopen Library* itself by using a timer or the timer interrupt function is called cyclically by another system timer.

Order Information

The *CANopen Driver Packages* are available as:

- reviewed hardware/system driver (DP 0565/xx) for typical hardware configurations
- freely selectable hardware driver CAN controller and CPU/OS combinations (DP/CPU 0566/xx, DP/CAN 0567/xx, DP/OS 0568/xx).

We recommend the purchase order of pre-configured driver packages that in addition to the CPU and CAN driver contain ready to use examples. These have been prepared with project files for the compilers and configurations we use.

For application of the *CANopen Library* with operating systems like Windows™ or LINUX™ ***port*** provides drivers for active and passive PC cards, parallel dongles, PCMCIA cards, USB interface and external Ethernet to CAN converter *(EtherCAN)*.

Predefined Driver Packages

Order Code	Name of System
0565/06	**DP C505C** microMODUL-505C/plain (*Phytec*) (on Request/8051 + 82527)[1]
0565/05	**DP C515C** miniMODUL-515C/plain (*Phytec*) (on Request/8051 + 82527)[1]
0565/03	**DP C167C** for miniMODUL-167C/plain (*Phytec*) (C166 + 82527)[1]
0565/34	**DP XC164CS DevBoard** (XC166 + TwinCAN)
0565/10	**DP Atmel T89C51CC01/02/03** (8051 + CANary)
0565/35	**DP Atmel AT90CAN128** (AVR + CANary)
0565/40	**DP Atmel AT91SAM7A2** (ARM7 with internal CAN)
0565/47	**DP Atmel AT91SAM7A3** (ARM7 with internal CAN)
0565/18	**DP STMicroelectronics ST72561** (with BeCAN)
0565/25	**DP STMicroelectronics STR730** (with C_CAN)
0565/33	**DP Silicon Laboratories C8051F040 DevBoard** (8051 + C_CAN)
0565/09	**DP Freescale HC908** (HC08 + msCAN)
0565/07	**DP Freescale HCS12(X)** for MC9S12(X) family (HCS12(X) + msCAN)
0565/48	**DP Freescale MC56F8323** (56800E + FlexCAN)[1]
0565/08	**DP Freescale DSP56F807** (DSP56800 + msCAN)
0565/38	**DP Freescale MPC565/566** (MPC5xx + TouCAN)[1]
0565/36	**DP Coldfire 5282** (with internal FlexCAN)[1]
0565/30	**DP Fujitsu MB90F543** (with internal Fujitsu CAN)
0565/66	**DP Fujitsu MB96F430** (with C_CAN)
0565/31	**DP TMS320LF2407 for eZdsp Board** (TMS320 + internal CAN)
0565/32	**DP TMS320F2808/2812 for eZdsp Board** (TMS320 + eCAN)
0565/26	**DP TMS470R1B1M** (ARM7 with HECC)
0565/19	**DP Microchip dsPIC33F256** (dsPIC33F with ECAN)
0565/28	**DP Microchip dsPIC24F256** (dsPIC24F with ECAN)

Order Code	Name of System
0565/37	**DP Microchip dsPIC30F6011/12/13/14** (dsPIC30F60xx with CAN)
0565/43	**DP Microchip PIC18F2680** (PIC18F2x8x with ECAN™)
0565/44	**DP Microchip MCP2515** (with SPI-Interface,example for XC164)
0565/39	**DP NXP LPC2129** (with internal CAN)
0565/46	**DP Beck IPC@Chip 1x3** (with internal CAN)
0565/49	**DP NEC PD70F3231** (V850/ES-FE2 with aFCAN)[1]
0565/21	**DP Renesas M16C29** (with internal CAN)
0565/22	**DP Renesas M32C/87** (with internal CAN)
0565/24	**DP netX** (with internal CAN)
0565/27	**DP DSTni-EX** (DSTni System-on-a-Chip)
0565/13	**DP CPC-Windows™** CPC CANopen driver for Windows™ (*EMS Wünsche*) e.g. useable for passive CPC-PCI card, CPC-USB Interface
0565/16	**DP Level-X Windows™** CANopen Treiber for Windows™, (*I+ME ACTIA*) e.g. useable for active PCI card PCI-IntelliCAN, USB-XS Interface, PC/104-board
0565/15	**DP Peak-light Windows™** CANopen driver for PCAN-light Interface (*PEAK-Service GmbH*), (on request)
0565/17	**DP CAN-AC2 Windows™** CANcard API compatible hardware (*Softing*); Driver for Win9x,NT; (on request)
0565/29	**DP Arcom Boards Windows™-CE** with AIM104-CAN-Module
0565/01	**DP AT-CAN-MINI** (***port***) CAN PC-Card AT-CAN-MINI-ISA MS-DOS (***port***)
0565/50	**DP can4linux** CANopen driver for can4linux compatible hardware under LINUX™
0565/14	**DP CPC-Linux** CPC CANopen driver for LINUX™, (*EMS Wünsche*) useable for passive CPC-PCI card, CPC-USB Interface
0565/51	**DP CAN232 Linux** Lawicel CAN232 driver for LINUX™
0565/52	**DP CAN232 Windows™** Lawicel CAN232 driver for Windows™

Order Code	Name of System
0565/60	**DP esdntcan4linux** esd CANopen driver for LINUX™ useable for cards with esdnt Interface

Adaptable Drivers *CAN driver*

Order Code	CAN controller type
0566/01	**DP/CAN Philips SJA1000**
0566/03	**DP/CAN Intel 82527**[1]
0566/04	**DP/CAN Siemens SAE 80C90/91** (on request)
0566/06	**DP/CAN Freescale TouCAN**[1]
0566/07	**DP/CAN Freescale msCAN** msCAN8, msCAN12, msCAN12/2
0566/08	**DP/CAN Fujitsu CAN** (16LX serie)
0566/09	**DP/CAN Atmel CANary**
0566/11	**DP/CAN Mitsubishi M16C/6N**
0566/13	**DP/CAN TI eCAN** (e.g. TMS320F28xx serie)
0566/14	**DP/CAN Bosch C_CAN** (e.g. Silabs C8051F040, Fujitsu MB91xxx, ST STR730)
0566/15	**DP/CAN Infineon TwinCAN**
0566/16	**DP/CAN Freescale FlexCAN**[1]
0566/18	**DP/CAN Atmel AT91SAM7A2**
0566/19	**DP/CAN Philips LPC21xx**
0566/20	**DP/CAN Microchip MCP2515**
0566/22	**DP/CAN Atmel AT91SAM7A3/SAM7X**
0566/23	**DP/CAN NEC aFCAN**[1]
0566/24	**DP/CAN TI HECC** (e.g. TMS470 serie)

CPU driver

Order Code	CPU type
0567/01	**DP/CPU Freescale MC683xx** (on request)
0567/02	**DP/CPU Freescale MC680x0** (on request)
0567/05	**DP/CPU Infineon C16x**
0567/06	**DP/CPU TI TMS320F2000**
0567/07	**DP/CPU Fujitsu MB90F54x,49x**
0567/08	**DP/CPU Freescale MPC823e**
0567/09	**DP/CPU Atmel T89C51CC0x**
0567/10	**DP/CPU Dallas 80C390/400**
0567/11	**DP/CPU Mitsubishi M16C/6N**
0567/12	**DP/CPU Freescale DSP 56F800**
0567/13	**DP/CPU Freescale HCS12(X)(X)**
0567/14	**DP/CPU SiLabs 8051**
0567/15	**DP/CPU Freescale HC08**
0567/16	**DP/CPU Infineon XC166**
0567/17	**DP/CPU MicroChip dsPIC30F60xx**
0567/18	**DP/CPU Atmel AT91SAM7A2**
0567/19	**DP/CPU Philips LPC21xx**
0567/20	**DP/CPU Freescale Coldfire**
0567/21	**DP/CPU Atmel AVR AT90CAN128**
0567/22	**DP/CPU Microchip PIC18F2x8x**

Order Code	CPU type
0567/23	**DP/CPU Atmel AT91SAM7A3**
0567/24	**DP/CPU Freescale 56800E**
0567/25	**DP/CPU NEC PD70F3231**
0567/26	**DP/CPU TI TMS470B1M**
0567/27	**DP/CPU Microchip dsPIC33F**
0567/28	**DP/CPU Microchip PIC24H**

Operation system

Order Code	OS type
0568/05	**DP/OS Windows 2000/XP**
0568/10	**DP/OS VxWorks**
0568/15	**DP/OS RTX-51**
0568/16	**DP/OS RTX-166**
0568/20	**DP/OS LINUX**
0568/xx	**DP/OS embOS** (on request)
0568/xx	**DP/OS FreeRTOS** (on request)

The supported number of drivers is constantly extended. If it happens to happen that your hardware is not yet listed we would be happy to provide support for developing your own driver for the *CANopen Library*.

1) hardware dependent deviations from standard functionality

CANopen Evaluation Software

Overview

The *CANopen Evaluation Software* contains all necessary software components to create and to test full functioning CANopen devices on different hardware targets. It is useful to explore of all the advantages and power of the *CANopen Library* and the associated tool chain.

To have a fast start ready configured examples and detailed documentation of the *CANopen Library* and software tools are provided on CD.

Description

The *CANopen Evaluation Software* makes it possible to learn more about the *CANopen Library* and the software tools without a great investment in time and money. Available are:

- the object code for the *CANopen Library*
- the object code for the CAN driver
- application examples in C source code
- the project files for the compiler
- the User and Reference Manual
- design and device commissioning tools

The following communication services are supported by the object code library:

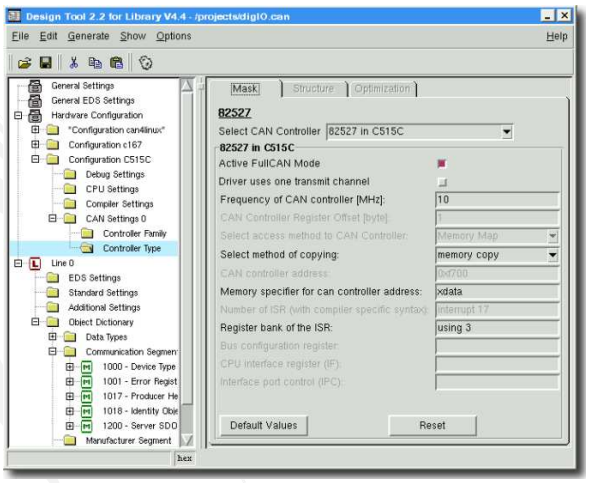
- Minimum Boot Up
- NMT services
- Service Data Object (SDO)
- Process Data Object (PDO)
- Heartbeat

All the examples are designed to use only internal resources of the respective micro controllers. Therefore they mostly can be used independent of any specific evaluation board.

All examples are provided in C source code to be able to adapt it to your needs. It is possible to:

- extend the object dictionary
- change the content of the PDO mapping
- respond on special CANopen events (like writing values to the object dictionary, or changing the NMT state)

Examples can be configured using the *CANopen Design Tool*, to carry out desired changes on the object dictionary or to adapt it to the hardware.


Design Tool 2.2 for Library V4.4 - /projects/digIO.can
File Edit Generate Show Options Help
Mask Structure Optimization
82527
Select CAN Controller 82527 in C515C
Active FullCAN Mode
Driver uses one transmit channel
Frequency of CAN controller [MHz]: 10
Select method of copying: memory copy
Memory specifier for can controller address: xdata
Register bank of the ISR: using 3
Default Values
Reset

Supported CPU's

port makes available the *CANopen Evaluation Software* for the following microprocessor families with internal CAN controllers

- Atmel T89C51CC01
- SiLabs C8051F040
- Infineon C505C
- Infineon C515C
- Infineon C167CR
- Infineon XC164
- Fujitsu MB90F497
- Fujitsu MB90F543
- Freescale HC908 68HC908GZ60
- Freescale HCS12 MC9S12DG128B
- Freescale DSP 56807
- Texas Instruments TMS320LF2407
- Texas Instruments TMS320F2812

Using the *CANopen Evaluation Software* for related controllers is mostly possible without problems. In general this means adaption of the initialization software and linking it to the correct addresses.

Compiler

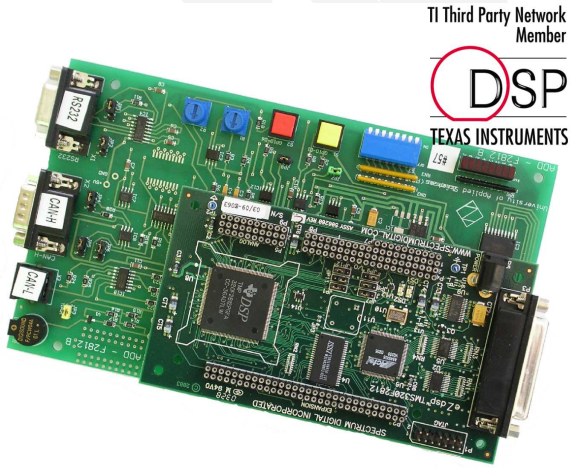
The projects on CD were created by the following compilers:

- Atmel: µVision2 (Keil-C51)
- SiLabs: µVision2 (Keil-C51)
- Fujitsu: Softune Workbench
- Infineon family: µVision2 (Keil-C51, Keil-C166)
- Infineon C166, XC166 family: Tasking C166
- Freescale family: Code Warrior and Cosmic
- TI signal processors TMS320F240x: TI Code Composer
- TI signal processors TMS320F28xx: TI Code Composer Studio

Hardware Targets

Typical hardware targets of different manufacturers are supported:

- the modules produced by Phytec
- demo board of Atmel
- demo board of SiLabs (Cygnal)
- DIMM-Module of Graf Syteco with Fujitsu CPU
- eZdsp system kit of Spectrum Digital with Texas Instruments DSP's
- Demo boards from Freescale



eZdsp TMS320F2812 Board with Zwickau extension

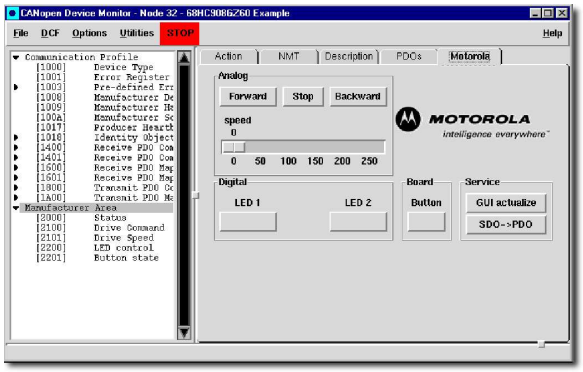
Restrictions

Compared with the full supported standard *CANopen Library* only some minor restrictions exist. These restrictions are mainly to reduce the memory requirements for the Library and the examples and supporting applications that can be built using only the internal resources provided by some processors. In case of the C515C both application and library do only use 1 K RAM. Furthermore the number of entries in the object dictionary is limited and the application restarts after 60 minutes.

Tools

port provides a complete tool chain for the development of CANopen devices. The following tools are available as demo version on *Evaluation Software*:

- Demo Version of the *CANopen Design Tool*
- Demo Version of the *CANopen Device Monitor*
- Demo Version of the CAN-RE***port***



custom designed mask for the M68HC908GZ60 example using CANopen Device Monitor

and a complete documentation

- *CANopen Library* User and Reference Manual as PDF- and HTML-format files
- *CANopen Design Tool* User manual
- *CANopen Device Monitor* User Manual
- CAN-RE***port*** User Manual

The tools contained on CD should give an impression how the device developer is supported at design, implementation and test of CANopen devices.

Therefore some of the examples are instrumented with additional scripts to enhance the tools with custom functionality.

License

The license coming with the evaluation software allows the usage of the provided software for evaluation purposes for the all products covered by the license. Foremost if the software should be used in commercial products the user of the evaluation CANopen software is engaged to contact the technical support at ***port*** to obtain a fully licensed CANopen library product.

Ordering Information

0564/39 CANopen Evaluation-Software

Functional demo versions of the software tools are available for download on *www.canopen-tools.com*.

CANopen Bootloader

2

Modern device designs need enormous flexibility in hard- and software. Today hardware is equipped with plenty of resources in order to allow software changes later on. Software itself has to provide means to enable an update/upgrade of the current firmware. At this point in-system-programming offers the greatest possibilities, and thus a change of the software too.

The desire for this feature also exists for CANopen devices and it is easy to implement. With the SDO transfer CANopen provides all means to transmit large chunks of data that occurs with a software update.

This CANopen binary bootloader implements a minimal node in a CANopen network as specified in the CiA Draft Standard Proposal 302 "Framework for Programmable CANopen Devices". It allows to use regular CANopen master software or configuration tools to download new firmware into the user flash code memory.

First implementations for the TMS320F28x DSPs of Texas Instruments, Fujitsu MB90xx and the XC164CS family of Infineon, are available.

The free version, available as binary FLASH image, has only few limitations:

- use of predefined bit rates. Configuration is done through writing the BTR values to FLASH. (When purchasing the source code the software can be adopted to manufacturer specific methods to determine bitrates)
- The bootloader has to be linked on a fixed address within the user code flash memory.
- Some object directory entries are linked statically and cannot be changed.

The full version without restrictions can be obtained from ***port***-GmbH via licensing agreement and doesn't have the restrictions mentioned above. For details please contact service@port.de.

Hardware Requirements

Through the use of the hardware independent *CANopen Sourcecode Library* of ***port*** the bootloader can be used on all supported target platforms. Solely the flash routines have to be adopted. Please inquire for the availability of the targets.

As a client a PC interface hardware (USB-CAN CPC-USB or USB-XS, PCI, ISA) or a gateway according to the CiA 309-3 (EtherCAN) is needed.

Software Requirements

A CANopen software able to do SDO domain transfers and supported by a CAN interface board is needed. A download program runnable under LINUX™ or Windows™ as well as some other tools, e.g. to generate a CRC checksum, can be downloaded from ***port***'s web-site.

Installation and Setup

In the upper area of the bootloader the configuration areais located. Adjust the values of the configuration according your requirements with a hex-editor.

```
struct {
/* 0xC0'3FE0 */   UNSIGNED32 vendor_id;
/* 0xC0'3FE4 */   UNSIGNED32 product_code;
/* 0xC0'3FE8 */   UNSIGNED32 revision_number;
/* 0xC0'3FEC */   UNSIGNED32 serial_number;
/* 0xC0'3FF0 */   UNSIGNED8 nodeid;
/* 0xC0'3FF1 */   UNSIGNED8 btr0;
/* 0xC0'3FF2 */   UNSIGNED8 btr1;
} USER_CONFIGURATION_T;
```

After filling in the configuration data you should now be able to talk to the node using your CANopen configuration tool.

This is the list of implemented Object Dictionary entries:

Index	Sub	Data Type	Acc	Function
0x1000	0x00	u32	co	Device Type
0x1001	0x00	u8	ro	Error Register
				Error Field
0x1003	0x00	u8	ro	Number of Entries
0x1003	0x01	u32	ro	Standard Error Field
0x1008	0x00	vs	co	Device Name
0x1009	0x00	vs	co	Hardware Version
0x100a	0x00	vs	co	Software Version
0x1014	0x00	u32	ro	COB-ID EMCY
0x1017	0x00	u32	rw	Producer HB time
				Identity Object
0x1018	0x00	u8	co	Number of Entries
0x1018	0x01	u32	co	Vendor ID
0x1018	0x02	u32	co	Product Code
0x1018	0x03	u32	co	Revision Number
0x1018	0x04	u32	co	Serial Number
0x1F50	0x00	u8	ro	Number of Entries
0x1F50	0x01	domain	wo	Download Program Area
0x1F51	0x00	u8	ro	Number of Entries
0x1F51	0x01	u8	wo	Program Control

Note: The memory optimized Bootloader does not contain all optional objects and does not implement services not mandatory.

Device Type
: Contains 0xFFFF as Bootloader profile id in the lower part of the "Device Profile" area. In the upper part there is the identification of the download capabilities, e.g. possible data formats.

Error Register
: Always 0. An error during programming is not indicated here but rather through an "SDO Abort" message or an "EMCY" message.

Standard Error Field
: Contains the error code of the lately sent EMCY

Manufacturer Device Name

Manufacturer Hardware Version

Manufacturer Software Version

Heart Beat Producer Time
: Period for sending heartbeat messages. This index is configurable via CANopen.

Identity Object
: *Vendor ID* ***port*** . Reads back 0x0000'0034, ***port***'s official vendor ID assigned by the CiA.

Product Code
: At time of delivery the object contains a number of ***port***-GmbH . This value can be configured in the configuration area.

Download Program Area
: Use this entry to download your Intel hex file with the programming data. Please note that the download area is located on a fixed address. Download file format is binary, with an application header prepended.

Program Control
: Writing 0x01 into this entry will start the loaded application. Writing 0x04 into this entry will "ERASE" the application FLASH area.

Implementation Details and Application Requirements - XC164CS

The bootloader is linked at address 0xC0'0000. The user code flash area begins at 0xC0'4000. The first 128 bytes, 0xC0'4000 to 0xC0'407F, are used for administration. It contains information about length of the application and a cyclic redundancy checksum.

```
struct {
  UNSIGNED32  length;
  UNSIGNED16  crc;
  UNSIGNED16  applicationType; /* reserved */
  UNSIGNED32  addr;            /* reserved */
} APPLICATION_HEADER_T;
```

This area is filled with 0x00. The application has to be linked to the flash area 0xC0'4080 to 0xC0'1FFF. The interrupt vector table should start at 0xC1'0000. The bootloader jumps at the start address 0xC1'0000.

EDS File

The file Bootloader.eds ⟨`http://www.port.de/eng/canprod/sw_bootloader.html`⟩ is a standard CANopen **Electronic Data Sheet** that can be used with a CANopen configuration tool, like the CANopen Device Monitor ⟨`http://www.port.de/engl/canprod/sw_monitor.html`⟩, to allow immediate access to the object dictionary entries.

Download

The binary and Intel-hex file of the bootloader and some tools can be downloaded from our Web Site.

cksum
Checksum program LINUX executable

cksum.exe
Checksum program Windows executable

downloader
LINUX command line Downloader; transfers a file using Domaintransfer

downloader.exe
Windows command line Downloader for EMS interface boards; transfers a file using Domaintransfer

Tools

There are two additional programmes to assist with the program download.

cksum produces a complete "ready to download file" with prepended application header from the binary image. This header contains information about length and checksum, which are stored in flash memory

Use cksum:

```
C:> cksum -C -O download.bin application.bin
```

Further information are given using the option **-h**.

downloader.exe executes the SDO domaintransfer as a client. This program takes the file as argument and transmits it. The bootloader writes the received programme into the flash memory.

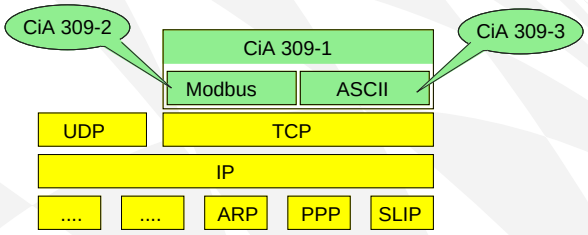
Ordering Information

0261/01	Bootloader Binary Infineon XC164
0261/02	Bootloader Binary Fujitsu MB90F5xx (on request)
0261/03	Bootloader Binary TMS320F2812
0261/05	Bootloader Binary TMS320F2808
0261/50	Generic Bootloader Source Code
0261/51	Bootloader Source Code Infineon XC164 (for Tasking Compiler)
0261/52	Bootloader Source Code Fujitsu MB90F5xx (on request)
0261/53	Bootloader Source Code TMS320F2812
0261/55	Bootloader Source Code Infineon XC164 (for Keil Compiler)
0261/56	Bootloader Source Code Philips LPC2129 for Keil Compiler
0261/57	Bootloader Source Code TMS320F2808
0261/60	Downloader Source Code

CANopen Gateway Server CiA 309-3

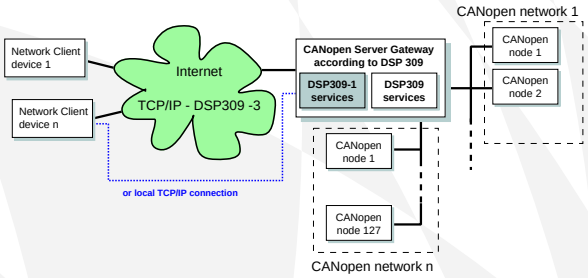
Overview

The subject *Ethernet access* down to the fieldbus layer is intensely discussed. The CANopen Gateway Server provides the solution of Ethernet access down to the fieldbus layer right now and relies on the current and field proven CANopen technology.



Description

With the specification CiA 309 the CiA provides a standardized solution to access a CANopen network through an Internet (TCP/IP) connection. The CANopen Gateway Server realizes a complete Class 3 CANopen Manager node.



It can operate on an arbitrary number of TCP/IP clients and CANopen networks. The CANopen CiA 301 services NMT, SDO, PDO, EMCY, Heartbeat and Nodeguarding are supported as described within the CiA 309-1.

With the SDO domain transfer services it can carry out up and downloads of firmware or any other data like images. Additionally it acts as LSS master and allows complete configuration of unconfigured devices.

The CANopen Gateway Server can be used as ready to use CANopen application to setup a graphical interface to access a CANopen network like it is done with the company's proprietary CANopen Device Monitor. Instead of a general purpose CANopen monitor interface complete user applications can be developed that provide maintenance and service menus to administrate or configure the CANopen nodes in CANopen networks.

Special purpose applications can be built with the source code of the CANopen Gateway Server. It allows the extension of the CiA 309-3 command interface to provide further functionality to access other devices that are connected via RS232/RS485 or parallel interfaces.

CAN-Interfaces

The usage of the CANopen Gateway Server requires a CAN interface. CAN interfaces are currently available for USB, ISA, PCI, parallel port, serial port and PC-104 or an Ethernet interface. The CANopen Server is available for the standalone Ethernet gateways EtherCAN of ***port*** and IGW900 of SSV.

EtherCAN IGW900 CPC-USB

System Requirements

Operating System:	Windows™ 2000, XP, LINUX™, Embedded systems
Processor:	any (>= 16 Bit CPU)
Software:	TCP/IP protocol stack or serial interface

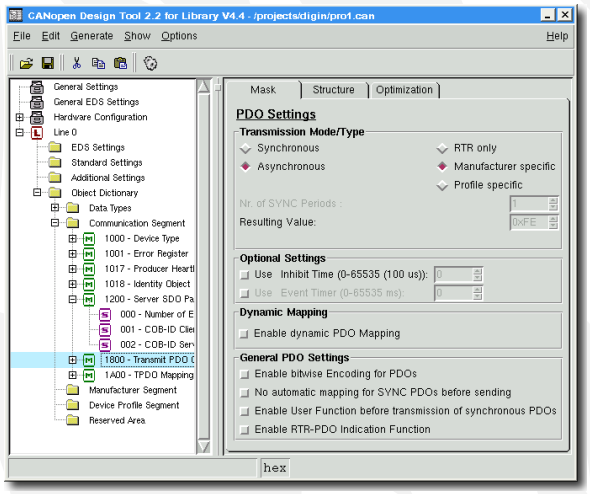
Ordering Information

0644/03	Source Code
0644/11	PC Software for Level-X
0644/17	PC Software for CPC-PCI
0644/18	PC Software for CPC-USB
0644/19	PC Software for Linux
0644/22	Software for Beck IPC@Chip SC1x3
0540/04	EtherCANopen standalone gateway
0850/03	IGW/900 standalone gateway

CANopen Design Tool

Overview

The *CANopen Design Tool* is a tool for rapid and cost-saving development of CANopen applications (devices). It generates an object dictionary and an initialization function in C-code, an Electronic Data Sheet and the documentation of the project automatically. Furthermore it simplifies the configuration of the *CANopen Library* and of the *CANopen Driver Packages*.



Description

With the *CANopen Design Tool* a powerful tool is available, which frees the developer from error-prone and repetitive tasks. It guarantees the consistency of the implemented functionality, device documentation and Electronic Data Sheet (EDS). By default the CANopen communication profiles CiA 301 and CiA 302 are delivered with the tool.

In addition to the CiA 301 database databases are provided for a large number of CANopen device profiles and for the CANopen communication profile CiA 304. The object dictionary produced by the *CANopen Design Tool* supports numerous options of the *CANopen Library* from ***port***, e.g. several CAN lines and segmented structuring. A tree representation of all implemented parameters and data facilitates the maintenance of the device software. The *CANopen Design Tool* eases the first step into the CANopen protocol and device development is accelerated.

It is a tool with which device databases can be produced and processed. These databases contain information, which describe the interface for the CANopen network of the device. These are all parameters, control and status information that are accessible over CAN. The CANopen protocol stores these values in the so-called object dictionary of CANopen devices.

The major task of the *CANopen Design Tool* is to administer and create these data in an object database and to create code for the *CANopen Library* from it. Furthermore the *CANopen Design Tool* is used to configure the *CANopen Driver Packages*. Within one project several hardware configurations can be managed. This allows a quick change between different targets during development.

CANopen Object dictionary

The generated file, which contains the object dictionary is included into the user modules. This ensures that each user variable can be accessed directly via its variable name as well as with its index and sub index. The produced object dictionary represents the data interface between library and user application.

CANopen Library Configuration

To configure the *CANopen Library* a header file is created which contains the hardware settings and the configuration of the used CANopen services. These services are derived automatically from the object dictionary. By including this file into the project only the necessary parts of the *CANopen Library* are compiled in order to save RAM and ROM resources.

CANopen Initialization

Likewise, the generated initialization file with initialization functions for the CANopen services is included into the application, too. So the individual services defined in the object dictionary are initialized in the application.

EDS File

A fourth result of the tool is the Electronic Data Sheet. The EDS belongs to the standard documents supplied with a CANopen device. It is an ASCII text file, which contains all relevant information about the

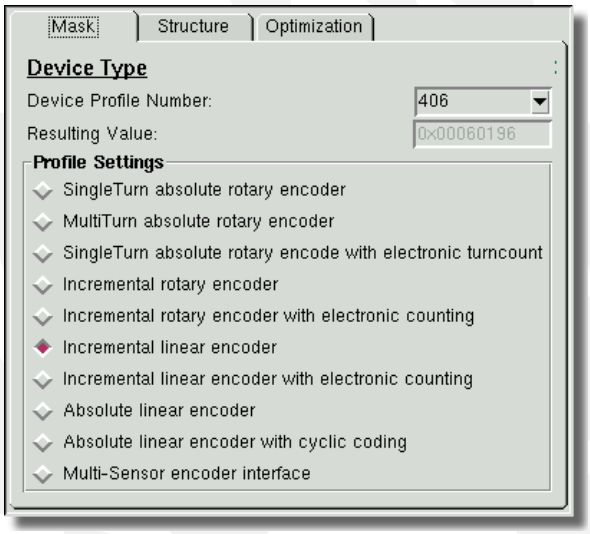
application (device). These files are used by configuration tools and control devices, in order to install CANopen devices into networks and configure them. In the future these files will be replaced by XML device descriptions according to ISO 15745. The *CANopen Design Tool* generates XML device description files already.

Device Documentation

The fifth output is a documentation of the implemented objects and of the device parameters. Each device parameter is tabular listed in html-format or as a text file. This documentation can serve as specification at the beginning of development and be brought into the user documentation, as well as into the marketing documents. Therefore writing twice is not necessary and the contents are always up to date.

Easy to Use

The objects of the application can be created by a few clicks. For the communication parameters input masks facilitate the creation of the CANopen service objects and the configuration of partially bit-coded values like e.g. PDO settings. Standardized objects based on CANopen device profiles can be added from pre-defined profile data bases.



Configuration mask for the object 0x1000

Generation

After an automatic check of the data for inconsistencies, missing entries or incompatible data types the following files are created:

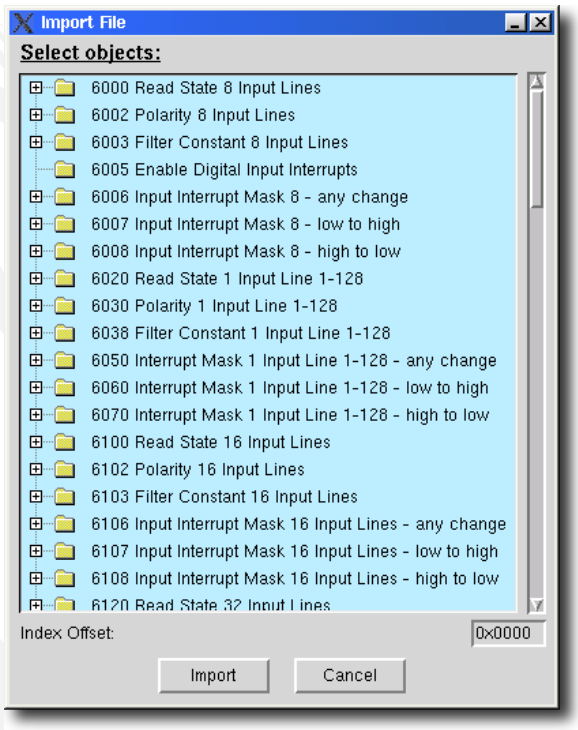
- the object dictionary (**objects.h**),
- the initialization file (**co_init.c**),
- the configuration file (**cal_conf.h**),
- the EDS files as EDS and XML file,
- and documentation files as HTML and text.

If the check fails, errors and warnings are written into the log file **generate.err**.

The results can be viewed with the internal file viewer with syntax highlighting.

Profile Databases

The profile databases contain the objects of the corresponding profile. Each object is configured according to the standard and is explained in detail. For every kind of object in a profile a prototype is included and this object can be cloned by copying or duplicating.


Import File
Select objects:
6000 Read State 8 Input Lines
6002 Polarity 8 Input Lines
6003 Filter Constant 8 Input Lines
6005 Enable Digital Input Interrupts
6006 Input Interrupt Mask 8 - any change
6007 Input Interrupt Mask 8 - low to high
6008 Input Interrupt Mask 8 - high to low
6020 Read State 1 Input Line 1-128
6030 Polarity 1 Input Line 1-128
6038 Filter Constant 1 Input Line 1-128
6050 Interrupt Mask 1 Input Line 1-128 - any change
6060 Interrupt Mask 1 Input Line 1-128 - low to high
6070 Interrupt Mask 1 Input Line 1-128 - high to low
6100 Read State 16 Input Lines
6102 Polarity 16 Input Lines
6103 Filter Constant 16 Input Lines
6106 Input Interrupt Mask 16 Input Lines - any change
6107 Input Interrupt Mask 16 Input Lines - low to high
6108 Input Interrupt Mask 16 Input Lines - high to low
6120 Read State 32 Input Lines
Index Offset:
0x0000
Import
Cancel

Import mask for a profile database

By importing these objects into the own project the development time can be reduced and the risks of a manual implementation are minimized.

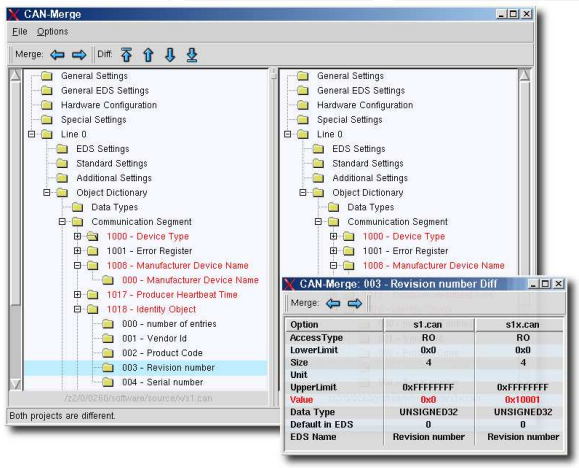
The following profile databases currently available:

- CiA 304 - Framework for Safety-Relevant Communication
- CiA 401 - Device Profile for Generic I/O Devices
- CiA 402 - Device Profile for Drives and Motion Control
- CiA 404 - Device Profile for Measuring Devices and Closed- Loop Controllers
- CiA 405 - Device Profile for IEC1131-3 Programmable Devices
- CiA 406 - Device Profile for Encoders
- CiA 410 - Device Profile for Incliometer
- CiA 415 - Device Profile for Road construction machinery
- CiA 418 - Device Profile for Battery Modules

- CiA 419 - Device Profile for Battery Charger
- CiA 417 - Application Profile for Lift Control Systems

CAN-Merge

The optionally available CAN-Merge PlugIn is a tool to compare and to merge *CANopen Design Tool* projects. A detailed presentation of each attribute of indices and sub indices and comfortable editing facilities simplify the handling of multiple projects. The integrated diff report generator generates expressive diff reports in XML.



CAN-Merge shows its strength especially when dealing with large projects or with large development teams.

Advantages

- acceleration of the development by avoidance of double work
- error prevention during creation of the object dictionary and EDS
- consistent documentation and EDS file to the respective device version
- reuse of data for similar devices
- supply of data for standardized CANopen device profiles
- optimized configuration of the *CANopen Library* according to the implemented CANopen services

System Environment

The *CANopen Design Tool* runs on PC's with

- Operating System: Windows™ 2000, XP, Vista and LINUX™
- Processor: Pentium III or later
- RAM: 256 MByte
- Harddisk Space: 20 MByte

Ordering Information

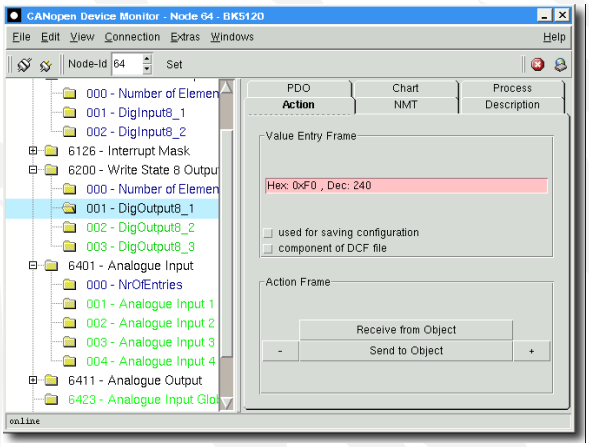
0640/05	CANopen-DT-W-Light (Light Version Windows™)
0640/06	CANopen-DT-L-Light (Light Version LINUX™)
0640/10	CANopen-DT-W (Full Version Windows™)
0640/20	CANopen-DT-L (Full Version LINUX™)
0640/11	CANopen-DT-P401 profile database for CiA 401
0640/12	CANopen-DT-P402 profile database for CiA 402
0640/13	CANopen-DT-P406 profile database for CiA 406
0640/17	CANopen-DT-P304 profile database for CiA 304
0640/21	CANopen-DT-P417 profile database for CiA 417
0640/23	CANopen DT-P404 profile database for CiA 404
0640/24	CANopen DT-P405 profile database for CiA 405
0640/26	CANopen DT-P410 profile database for CiA 410
0640/28	CANopen DT-P418 profile database for CiA 418
0640/29	CANopen DT-P419 profile database for CiA 419
0640/50	CANopen-EDS2XML-W EDS-to-XML converter Windows™
0640/51	CANopen-EDS2XML-L EDS-to-XML converter LINUX™
0640/49	CANopen-DT-CAN-Merge CAN-Merge PlugIn

Functional demo versions of the software tools are available for download on *www.canopen-tools.com*.

CANopen Device Monitor

Overview

The *CANopen Device Monitor* is a tool for the graphical inspection and configuration of CANopen devices in a CANopen network. The embedded scripting ability provides access to the implemented CANopen services and makes it possible to write test or control applications with a minimum of effort.



The information on how to display the device object dictionary is obtained from the EDS file of the device (EDS - Electronic Data Sheet, text- or XML format according to ISO 15745) or can be scanned from the device directly. Special device configurations can be loaded, stored and scanned from DCF files. Additionally, configurations of complete networks can be saved in project files.

Description

The program's graphical user interface consists of

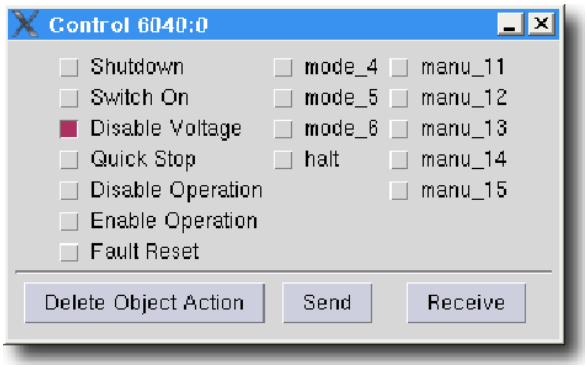
- a hierarchical tree representation of the object dictionary
- various action fields
- an optional command console interface

Displaying all implemented objects as a tree facilitates the configuration by hand.

Devices with up to 8 device profiles are supported.

The `tab Action` is used for reading and writing device objects with CANopen SDO transfer messages.

Optionally a slider with a user defined value range can be assigned to every object to achieve a more useful configuration. For each object the meaning of each bit can be displayed by special bit boxes. There are already prepared bit boxes for special objects like the CiA 402 control word.



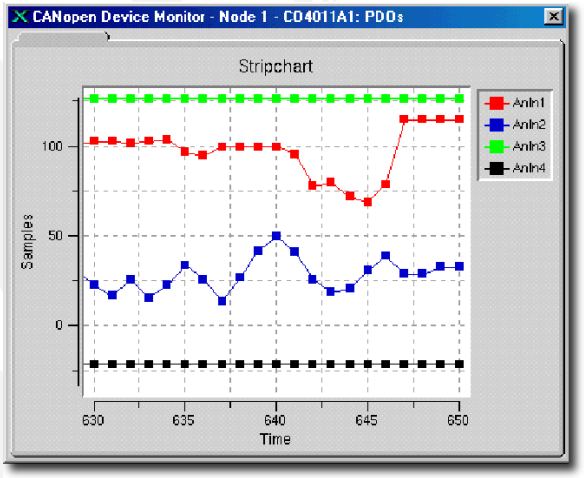
Values of manually configured device objects can be stored to a file and reloaded into the device.

The `NMT tab` makes several fields available for the execution of NMT services like *Start All*, *Start Node*, *Preop All*, *Preop Node* and *Scan Network*. The device properties of the currently present nodes in the network are shown as well.

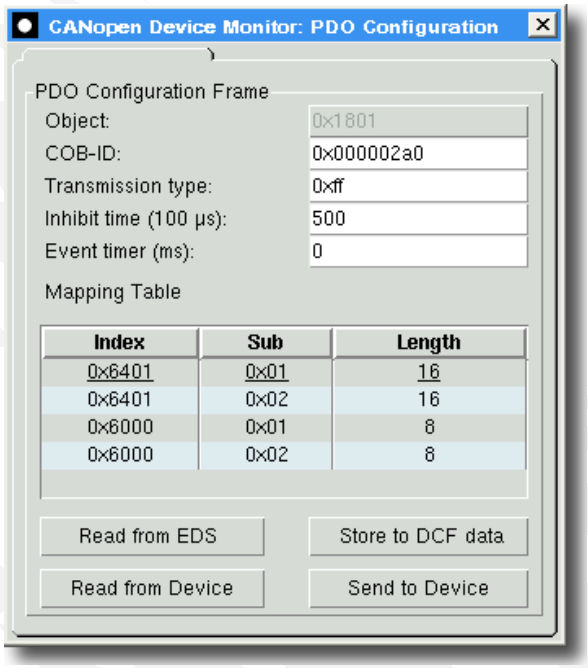
A graphical network overview displays all devices in a CANopen network.

Network configurations can be can be saved in project files.

Under the `tab Description` an object description of the currently selected object in the object tree is shown. The description is taken from a text file which is created by the *CANopen Design Tool* or manually written with a common text editor.



The integrated process image summarizes the most important process variables of the network. The data can be received via SDO and PDO. The `PDO tab` allows the simple configuration of the PDO-Mapping by Drag&Drop. Additionally PDOs can be sent and received and data from received PDOs can be plotted in a stripchart with one click.



PDO configuration mask

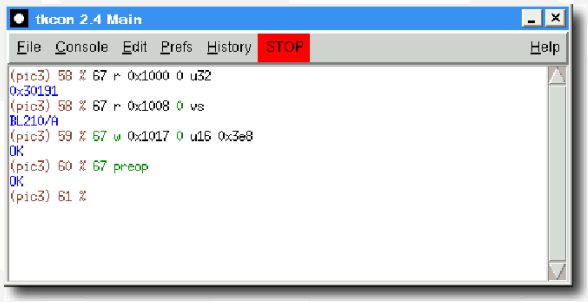
`Additional tabs` are available for further graphical representations of CAN data and can be defined by the user.

Extensions to control drives according to CiA 402, a CANopen-Safety-Plugin (CiA 304) and a LSS-module (CiA 305) are additional available.

The CAN communication can be logged, saved and evaluated afterwards.

The console is an extra feature. With the console an interactive access to the CANopen network is possible by using commands for:

- SDO and PDO transfers
- network management services
- Heartbeat and Node Guarding
- Time Stamp, SYNC, Emergency


tkcon 2.4 Main
File Console Edit Prefs History STOP Help
(pic3) 58 % 67 r 0x1000 0 u32
0x30191
(pic3) 59 % 67 r 0x1008 0 vs
BL210/A
(pic3) 59 % 67 w 0x1017 0 u16 0x3e8
OK
(pic3) 60 % 67 preop
OK
(pic3) 61 %

Complete sequences of commands can be combined to procedures by the embedded scripting language. The scripting language is available for different operating systems.

By using the *CANopen Library* and the *CANopen Design Tool* from ***port*** the developer gets a closed tool chain for creating CANopen compatible devices.

Through the scripting language access to elements of the graphical interface is also available. The graphical representation of values or test results is therefore application specific extensible.

port GmbH can perform special adaptions to the GUI or behavior to meet CANopen device characteristics for service or assembly testing.

CAN-Interfaces

The usage of the *CANopen Device Monitor* requires a CAN interface. CAN interfaces are currently available for USB, ISA, PCI, parallel port, serial port and PC-104 or an Ethernet interface like the standalone CANopen-Gateway-Server EtherCAN of ***port***.

System Requirements

Operating System:	Windows™ 2000, XP, Vista LINUX™
Processor:	Pentium IV or later
RAM:	256 MByte
Harddisk Space:	>20 MByte available

Ordering Information

0642/10	CANopen-CDM-W (Full Version, Windows™)
0642/20	CANopen-CDM-L (Full Version, LINUX™)
0642/11	CDM-CiA304 Safety Extension
0642/12	CDM-CiA305 LSS Extension
0642/13	CDM-CiA402 Drives Extension

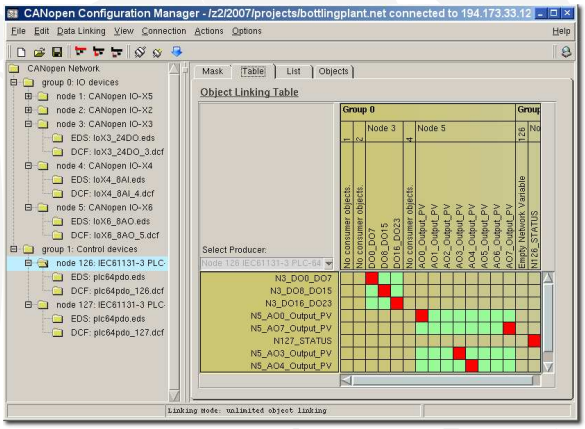
Functional demo versions of the software tools are available for download on *www.canopen-tools.com*.

CANopen Configuration Manager

Overview

Large CANopen networks with several devices and a multiplicity of inputs and outputs can have a complexity, that is hard to manage without powerful tools. The *CANopen Configuration Manager* is such a tool that simplifies the configuration of CANopen networks.

Using the *CANopen Configuration Manager* connections between devices and their configuration settings can be configured conveniently. So the PDO linking is simplified enormously.



Description

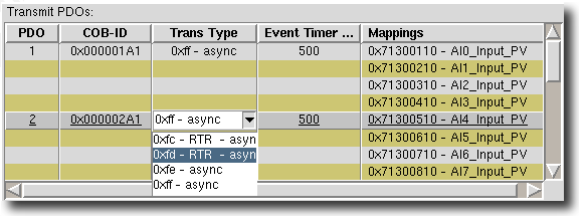
Each network is managed as a project within the *CANopen Configuration Manager*. Within a project the devices can be combined to different groups. Groups can be assigned arbitrarily according to device properties e.g. I/O devices, drives, according to node-ID ranges or with regard to their physical arrangement. Devices can be added to a group via their EDS files or a network scan (PC-CAN-Interface required). *CANopen Configuration Manager* project files can be reused by the *CANopen Device Monitor*.

Object links

Configuration of the object links is done by simply marking the objects shown in the table view. Internals of the CANopen protocol are hidden from the user and the presentation is application-specific. CANopen notations like node 3, object 0x6040, subindex 0x00 are replaced by the more descriptive parameter name from the EDS file, e.g. statusword. The selection of invalid combinations of application variables due to incompatible datatypes is not possible and so errors are eliminated in the preliminary stages of the configuration. Further attributes like priorities or SYNC intervals can be assigned to the links. These attributes are evaluated by the linking algorithm.

PDO linking

Based on the object links the *CANopen Configuration Manager* computes the PDO mapping entries and sets the COB-IDs for the PDOs of each device. Thereby attributes and special cases like static mapping, constant COB-IDs or a restricted COB-ID range are considered. A direct configuration of the PDO parameter is also possible. The program ensures consistency of project settings and device settings.



direct PDO configuration

Timing Configuration

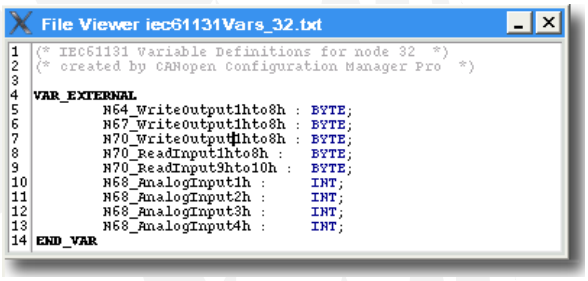
Further features of the *CANopen Configuration Manager* are its ability to configure and to coordinate timing parameters like e.g. SYNC or Heartbeat.

Object Configuration

All manufacturer or device profile objects of each device can be parametrized without restrictions.

CANopen-PLC support

Devices according to CiA 405 (CANopen-PLC) with dynamic network variables are supported by the *CANopen Configuration Manager*. When connecting an object of another device to the PLC, the network variable is created automatically. Thereby appropriate data types and attributes are assigned to the network variable.



IEC61131 Variable declarations

Program outputs

The following files are generated after completion of the configuration:

- DCF files
- Concise DCF files
- IEC61131 Variable declarations for PLCs,
- a detailed project documentation

Parameter download

When the computer is connected via a CAN interface to the network the complete configuration or only changes of a single node can be sent directly to the devices on the network. Similarly it is possible to read the configuration of a node or even a complete network. Additionally the *CANopen Configuration Manager* is able to control devices with NMT commands.

Advantages

- simplified PDO linking
- consistency of the configuration throughout the network
- direct parameter download possible
- support of CANopen-PLCs
- detailed project documentation
- reuse of the project files by *CANopen Device Monitor* is possible

CAN-Interfaces

The functionality of the parameter download requires a CAN interface. CAN interfaces are currently available for USB, ISA, PCI, parallel port, serial port and PC-104 or an Ethernet interface like the standalone CANopen-Gateway-Server EtherCAN of ***port***.

System Environment

The *CANopen Configuration Manager* runs on PC's with

- Operating System: Windows™ 2000, XP, Vista and LINUX™
- Processor: Pentium IV or later
- RAM: 512 MByte
- Harddisk Space: >24 MByte available

Ordering Information

0545/10	CANopen-CCMbasic-W (Windows™)
0545/12	CANopen-CCMpro-W (Windows™)
0545/20	CANopen-CCMbasic-L (LINUX™)
0545/22	CANopen-CCMpro-L (LINUX™)

Functional demo versions of the software tools are available for download on *www.canopen-tools.com*.

CANopen LINUX Starter Kit

Overview

For the constantly increasing interest in LINUX™ in industrial automation we take this into account by providing the *CANopen LINUX Starter Kit*. It provides the necessary hard- and software for developing *CANopen Library* applications for LINUX™.

Description

The *CANopen Starter Kit* enables the device developer to explore the capabilities and the advantages of the *CANopen Library*. With the prepared and modificable examples the developer can estimate the effort needed to create the own applications.
The starter kit contains a precompiled static library, which itself uses **glibc** functions, and enfolds the following CANopen services:

- NMT Slave
- SDO Server
  (more than one server SDO can be implemented)
  - ☐ segmented data transfer is possible
- PDO Producer
  - ☐ no dynamic PDO mapping
  - ☐ no SYNC PDOs
  - ☐ no bit wise mapping
- PDO Consumer
  - ☐ no dynamic PDO mapping
  - ☐ no SYNC PDOs
  - ☐ no bit wise mapping
- Heartbeat Producer
- object dictionary implementation

All applications generated with the starter kit library will be finished after half an hour run time.

Usage of the *CANopen Library* can be tested with prepared projects, which are available in source code with Makefiles. To use and test immediately the created applications demo-versions of all software tools are available on the following web site: URL http://www.port.de/engl/shop.html

Hardware

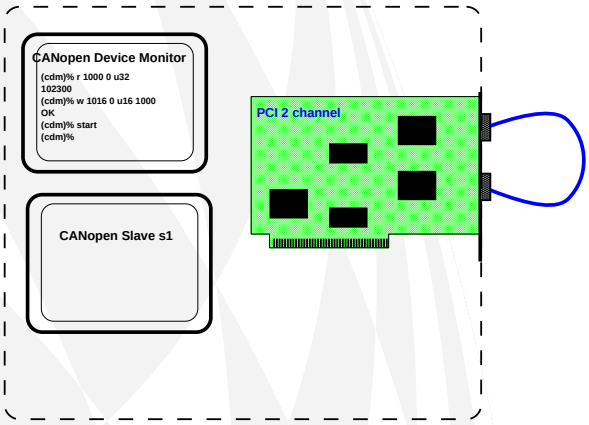
To allow a flexible choice of CAN interfaces, *can4linux* is used as LINUX™ CAN device driver. A very flexible compilation of interfaces is possible by this choice. The driver can be downloaded for free from the ***port*** web server ⟨http://www.port.de/engl/canprod/hw_can4linux_ext.html⟩

The following CAN interface boards are available:

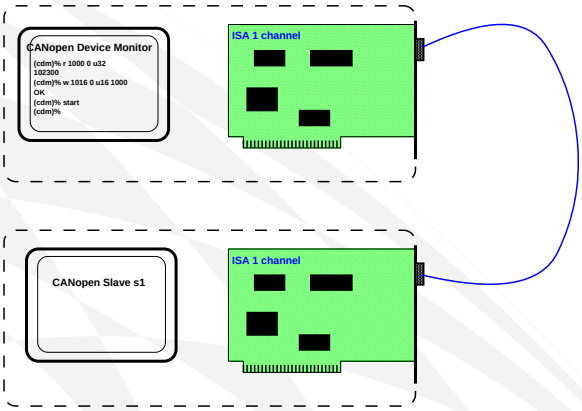
- AT-CAN-MINI,
  one CAN channel CAN ISA interface, galvanically decoupled
- CPC-PCI,
  one or two channel PCI board, galvanically decoupled on request
- EtherCAN,
  Ethernet TCP/IP to CAN Gateway

Configurations

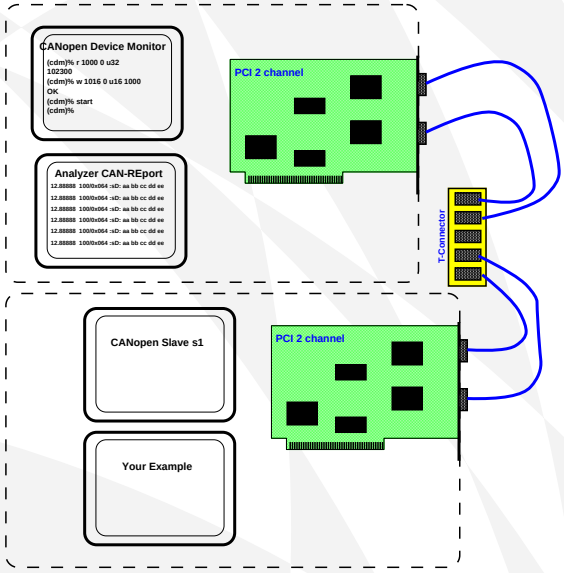
The following figures should introduce some of the possible configurations.



In the simplest case both, the example application and the *CANopen Device Monitor* are running on one PC. Each application needs one CAN interface. Either two channels of one card or two different CAN cards are used and are short connected.

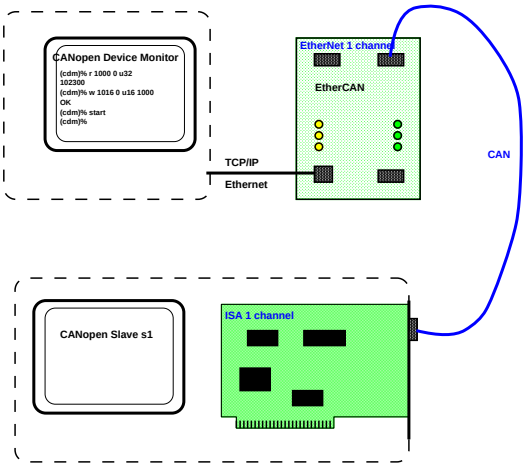


The next figure shows the same software configuration, this time each application running on its own PC, and each using it's own CAN interface.



When using two interface boards with two CAN channels each, you can use a second application besides the *CANopen Device Monitor* and the analyzer CAN-RE***port***.
Both interface boards can also be used in the same PC.

An interesting alternative solution to interface boards is offered with EtherCAN.



This interface can be used instead of local PC boards as interface for *CANopen Device Monitor* and CAN-RE***port***. The services provided by these tools are then available for other computers in the local network if connected by Ethernet.
Besides the software components, the delivery contains the User Manual and the Reference Manual of the *CANopen Library* and of the tools on CD-ROM.

System environment

The usage of the starterkit needs an Intel based PC system. The library can be provided on request for other systems. Furthermore a CAN interface, which is supported by *can4linux* has to be available.

Ordering information

0572/01 LCSK Base,
Base Slave Software Development Set; with Slave Sources, pre-compiled *CANopen Library*.

To select and order the needed starter kit components, please ask for the *CANopen LINUX Starter Kit* order form, containing also price information. We would be glad to send this form as PDF attachment by email.

Functional demo versions of the software tools are available for download on *www.canopen-tools.com*.

Software Components for DeviceNet

DeviceNet Slave Development Kit

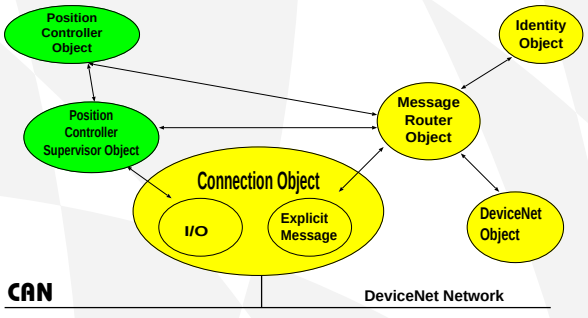
Overview

DeviceNet connectivity for your products.

DeviceNet is used in a wide range of industrial automation products and equipment. ***port*** provides a DeviceNet Slave Developers Kit (DSDK) that helps engineers to add the DeviceNet protocol quickly to a host of slave devices such as weld and robot controllers, sensors and medical equipment, and more.

The DSDK allows developers of embedded products to add UCMM (Unconnected Message Manager support) capable DeviceNet Slave functionality using field proven library. Designed for rapid integration into your DeviceNet products, the kit includes complete documentation, source code, and example application code for both an AMD 80188EM/SJA1000 CAN controller and the Infineon (Siemens) C167CR with internal CAN. All source code, with the exception of some microprocessor-specific startup code, is written in C.

Additional implementations for running the DeviceNet slave as an Linux application using the **can4linux** device driver and implementations using Texas Instruments DSPs TMS320F24x with internal CAN are available. Sample projects for these targets can be provided on request.



Features

DSDK features:

- Compiler support for the x188 and Infineon SABC167CR microcontroller
- Scalable for Group 2 Only or UCMM (Unconnected Message Manager support) capable devices
- Predefined group 2 connection set support for
  "Explicit Message"
  "Cyclic"
  "Polling"
  "Change of State"
  "Strobe Connections"
- UCMM (Unconnected Message Manager support) Explicit Message Server / Client
- Non-Volatile or DIP-Switch net parameter support
- Fragmentation of both Explicit and I/O messages
- Offline Connection Set
- Device Heartbeat Message producer
- Device Heartbeat Message consumer
- CAN controller supported:
  - Philips SJA1000 — stand alone CAN controller
  - Infineon SABC167CR — with integrated CAN controller
  - TMS320F2407 und TMS320F2812 — with integrated CAN controller
  - can4linux — Linux device driver
- supported Objects
  - Identity
  - DeviceNet
  - Connection
  - Assembly
  - Ack Handler
  - Message Router
  - Allen-Bradley Non-Volatile-Storage
  - Debug Template
  - Position Controller Supervisor Template
  - Position Controller Template

An adaptation of the DeviceNet software to meet the requirements of processors and CAN controllers not listed above can usually be accomplished within a few days using standard C code. We will gladly accomplish such an adaptation of the DeviceNet software on inquiry.

Position Control

For the development of a drive unit with a DeviceNet interface ***port*** provides a special add-on module. The Position Control Module bases on the objects Position Controller Supervisor and Position Controller and accelerates the device development by the handling of command and response messages and the drive-specific handshake for I/O messages. Routines for the services "Get Attribute Single" and "Set Attribute Single" make the drive configuration easier. Additional error indication functions supports the error handling. The Position Control Module allows the user to implement its own drive functionality, such as various profile types, special control algorithms, application-specific limit functions, standardized and individual modes.

Development Tools

The CAN-Analyzer CAN-RE***port*** is an efficient and versatile tool for analysis and starting of CAN-based networks like DeviceNet. The DeviceNet-specific interpretation of CAN messages is provided by a supplementary software module.

This module contains commands to establish communication channels and to read and write object attributes. It can be used to test the DeviceNet communication of devices built using the DSDK.

Ordering Information

0245/01	DeviceNet Slave Developers Kit — AMD80188EM/SJA1000
0245/02	DeviceNet Slave Developers Kit — C167CR
0245/03	DeviceNet Slave Developers Kit — LINUX™/can4linux
0245/04	DeviceNet Slave Developers Kit — CodeComposer TI TMS320F240x
0245/05	DeviceNet Slave Developers Kit — CodeComposer TI TMS320F2812 (on request)
0245/0x	add-on module Position Control
0570/11	DeviceNet-SDK Integration Support
0580/10	CAN-RE***port***-W (Vollversion Windows™)
0580/20	CAN-RE***port***-L (Vollversion LINUX™)
0580/12	CAN-RE***port*** DeviceNet Extension

Functional demo versions of the CAN-RE***port*** SW tools are available for download on http://www.canopen-tools.com

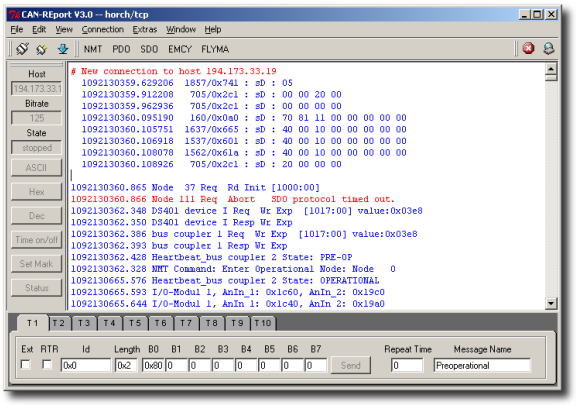
CAN Analyzer - CAN-RE*port*

Overview

The CAN-Analyzer CAN-RE***port*** is an efficient and versatile tool for analysis and commissioning of CAN-based networks according CAN 2.0A and 2.0B, like CANopen and DeviceNet.

The built-in scripting capability allows to use it universally at development and test of CAN devices besides the normal possibilities of displaying the received CAN messages. It is especially useful in the field of industrial CAN networking.

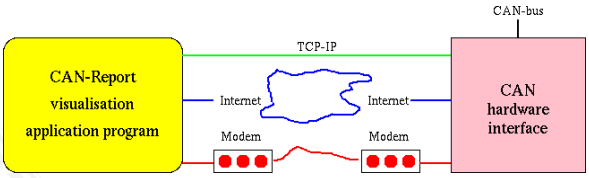
The separation of hardware-interface (CAN access) and visualization software allows the usage in TCP/IP networks.



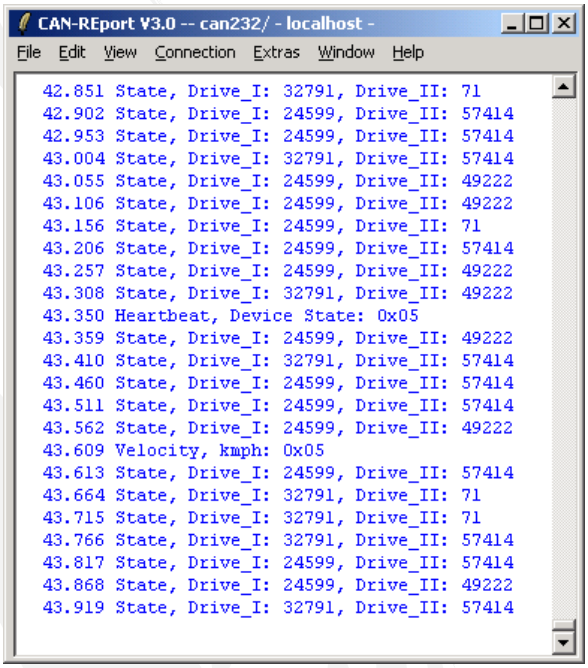
Description

In the standard equipment the CAN-RE***port*** already has a lot of efficient basic functions like different modes of online observation of the bus traffic, transmission of unique or cyclic CAN messages, trigger inclusive pre- and post-trigger logging as well as the recording of CAN messages and storing it to log files belong to this purpose. Log files can be loaded again and interpreted with activated extensions.

The analyzer CAN-RE***port*** consists of the CAN hardware interface and the visualization software. Both are connected as server and client by a standard TCP/IP network connection. This separation allows to use the CAN interface as a remote interface, ie. both parts can be located at different computer systems. Remote monitoring of CAN networks is possible without additional or modified software over LAN, dial or internet connections.



CAN interfaces are currently available for USB, ISA, PCI, PC-104, parallel port, serial port, like "Lawicel CAN232", or an Ethernet interface like the EtherCAN and IGW900 of ***port***.


CAN-REport V3.0 -- can232/ - localhost -
File Edit View Connection Extras Window Help
42.851 State, Drive_I: 32791, Drive_II: 71
42.902 State, Drive_I: 24599, Drive_II: 57414
42.953 State, Drive_I: 24599, Drive_II: 57414
43.004 State, Drive_I: 32791, Drive_II: 57414
43.055 State, Drive_I: 24599, Drive_II: 49222
43.106 State, Drive_I: 24599, Drive_II: 49222
43.156 State, Drive_I: 24599, Drive_II: 71
43.206 State, Drive_I: 24599, Drive_II: 57414
43.257 State, Drive_I: 24599, Drive_II: 49222
43.308 State, Drive_I: 32791, Drive_II: 49222
43.350 Heartbeat, Device State: 0x05
43.359 State, Drive_I: 24599, Drive_II: 49222
43.410 State, Drive_I: 32791, Drive_II: 57414
43.460 State, Drive_I: 24599, Drive_II: 57414
43.511 State, Drive_I: 24599, Drive_II: 57414
43.562 State, Drive_I: 24599, Drive_II: 49222
43.609 Velocity, kmph: 0x05
43.613 State, Drive_I: 24599, Drive_II: 57414
43.664 State, Drive_I: 32791, Drive_II: 71
43.715 State, Drive_I: 32791, Drive_II: 71
43.766 State, Drive_I: 32791, Drive_II: 57414
43.817 State, Drive_I: 24599, Drive_II: 57414
43.868 State, Drive_I: 24599, Drive_II: 49222
43.919 State, Drive_I: 32791, Drive_II: 57414

Plugins

Supplementary software modules provide extended functionality like service and protocol dependent interpretation of CAN messages. For the higher layer protocols CANopen , DeviceNet and J1939 a service dependent protocol interpretation is available.

The CANopen plugin displays CAN messages of the services:
SDO — PDO — NMT — EMCY — Flying Manager in separate windows. Moreover the display of the CANopen plugin can be tailored with the settings for "Node Names", "PDO Mapping" and "Emcy Mapping". This settings are saved project-oriented and can be loaded again. Sending of CANopen messages is supported by additional commands.

The DeviceNet plugin allows not only the interpretation of protocol messages but also the interactive input of requests on DeviceNet nodes, e.g. object attributes can be read and written.

Another additional module allows the analysis of redundant networks. With two CAN interfaces it is possible to analyze CAN messages simultaneously of two CAN networks. CANopen Maritime networks are important applications with redundant CANopen networks.

For logging and protocol analysis of J1939 networks the J1939 plugin is available. The data contents are presented in human readable text as bit and byte values depending on its meaning within the protocol. Besides the interpretation of received CAN messages J1939 specific messages can be sent with the scripting language Tcl/Tk. The PGN and data is passed to the function and the plugin puts together the CAN message and sends it.

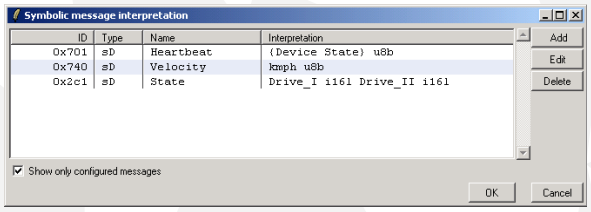
Programmability

For the interpretation of company internal protocols a plugin can be developed and loaded into CAN-RE*port*. Such a plugin is realized with the builtin scripting language Tcl/Tk and is simplified by provided functions. This mechanism allows it to simulate complete devices that utilize the CANopen, DeviceNet or a user specific protocol.

With the scripting language also elements of the graphical interface can be extended to match individual needs. It is possible to use provided or own graphical objects to create test and control applications. With the help of the multifaceted possibilities for displaying and reporting data test sequences can be carried out automatically even from a remote working place.

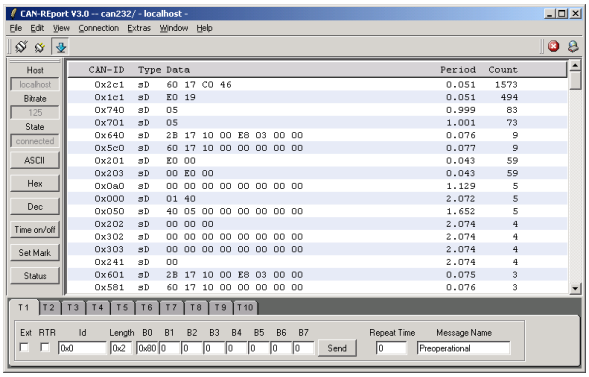
User Mapping

To get a simple and clear assignment of messages, they can be linked to a symbolic name. Not only every message but also the contents can be assigned to different identifiers, and interpreted according different data types. The integrated Tcl script support allows the very flexible extension of this mechanism.



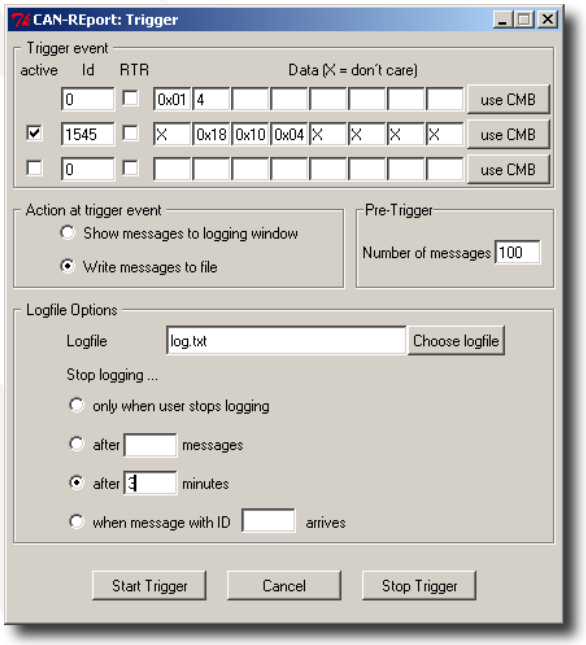
Object View

The object view is another representation form of CAN messages. Different than at trace mode, selected messages always remain visible and can be watched comfortably.



Trigger

Furthermore CAN-RE*port* provides the trigger functionality to wait for specific CAN messages and continue logging after the specific CAN message was received. The trigger dialog consists of the three parts *Trigger Event*, *Action at trigger event* and *Logfile options*. Up to three different CAN messages can be specified as trigger.



Interactivity

With the help of the integrated console convenient commands for accessing the CAN network interactively are available. These commands can be used in test scripts. Commands as `wr` for the sending of messages or `wait` for the synchronization with CAN messages on the network belong to this purpose.

With the following command sequence:

```
while { 1 } {
    wait 1799
    wr 0x100 0xaa [incr val]
}
```

the following message sequence can be observed:

The commands can be combined into sequences or procedures. All semantics of modern high level languages are available.

```
989563919.999016  1799/0x707 : D : 7f
989563921.024541  1799/0x707 : D : 7f
989563921.048045   256/0x100 : D : aa 01
989563922.046364  1799/0x707 : D : 7f
989563922.068331   256/0x100 : D : aa 02
989563923.027081  1799/0x707 : D : 7f
989563923.048192   256/0x100 : D : aa 03
989563924.047260  1799/0x707 : D : 7f
989563924.068092   256/0x100 : D : aa 04
989563925.071046  1799/0x707 : D : 7f
```

Particularly for commissioning and error analysis a precise time resolution of the received CAN messages at the network is necessary. The time represented by CAN-RE***port*** is influenced only by the used hardware. Time resolution up to 1 µs can be achieved by using CAN interface boards under the Linux operating system. The available recording functions allow to store the results of entire test runs, but also the content of logging windows, in separate files.
port can perform special adaptions to the GUI or behavior to meet special customer requirements for service or assembly testing.

Advantages

- CAN interfaces of different manufacturers
- support of remote monitoring of the CAN bus
- expandable for CANopen, DeviceNet, redundant CAN networks and propriety applications
- Trigger with pre- and post-trigger
- scriptability with Tcl/Tk (http://www.tcl.tk/)

System requirements

Operating System:	Windows™ 2000, XP, Vista LINUX™
Processor:	Pentium or later
RAM:	128 MByte
Harddisk Space:	10 MByte

Ordering Information

0580/10	CAN-RE***port***-W (Full version Windows™)
0580/20	CAN-RE***port***-L (Full version LINUX™)
0580/11	CAN-RE***port*** CANopen Plugin
0580/12	CAN-RE***port*** DeviceNet Plugin
0580/13	CAN-RE***port*** Maritime Plugin
0580/14	CAN-RE***port*** J1939 Plugin

Functional demo versions of the software tools are available for download on *www.canopen-tools.com*.

CAN Bus Tester

Highlights

- commissioning of CAN bus plants, Wiring test, module check
- service/maintenance of CAN bus plants
- troubleshooting and analysis of the physical bus characteristics
- development of CAN modules
- final testing in the production

Description

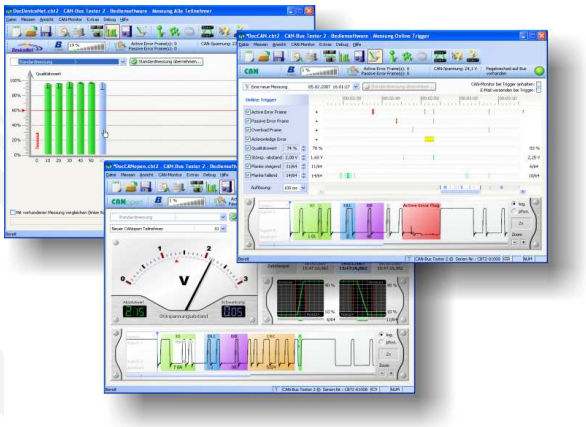
The *CAN-Bus Tester* permits a efficient start-up and a stable operation of CAN-Bus systems. Besides the protocol analyses you can identify problems on the physical bus level by measuring the disturbance-free voltage at run time.

A low disturbance-free voltage range is caused by bus problems at the physical level. A low disturbance-free voltage range indicates an insufficient resistance of your bus to EMC influences, for example.

A high disturbance-free voltage range, in contrast, represents an interferenceresistant system which saves you substantial costs and trouble caused by system failures.

The outstanding user-friendly system enables you to select different test modes for analyzing your bus with regard to a variety of characteristics, such as:

- overview of the disturbance-free voltage range of all telegrams
- more detailed analysis of a selectable telegram
- built in oszillocope function
- logger function for long-time analyses
- several trigger conditions for identifying faults and errors
- extended display of the IDs is available for DeviceNet or CANopen networks
- Wiring test. Determine line short-circuits, line breaks, the bus termination, the loop resistances of the CAN line and the CAN power supply line, and the overall line length.
- simple CAN monitor integrated



Technical Data

CAN Bus	ISO 11898-2 (High-Speed) 11 and 29 bit identifier (CAN2.0A and 2.0B)
CAN bit rates according to CAN type	10, 20, 50, 100, 125, 250, 500, 800, 1000 kBit/s
user-defined	5, 33.5, 62.5, 75, 83.3 kBit/s
bit rate detection	automatically
ID identification	by ID scan
list of measurement	sort by type, place, time
signal-to-noise ratio	typ.- 0.75 ... 3.0 V,
	resolution 50 mV
online trigger	resolution (10ms - 1min) real time inspection for logical and physic. errors
bus monitoring	permanently with regard to statically recessive or dominant, respectively, or bus running
bus workload factor	permanent display of percentage value
firmware and FPGA configuration	update by USB

Technical Data

trigger output for oscilloscope	BNC socket, galvanically separated
connection to CAN bus	passive
bus connection design	2x 9 pin. SUB-D connector connector with 1:1 connection
PC connection	USB V1.1, galvanically separated
update	via USB
supply voltage	extra-low-voltage socket according to DIN 45323
supply voltage	9 ... 36V DC
power consumption	0.55 ... 0.15 A
size (l x w x h)	170 x 125 x 40 mm
weight	570g
enclosure	aluminum box
safety/ protection	EN 60950-1/ IP 20 according to EN 60529
EMC	EN61000-3-2, EN55022
interference resistance	EN 61000-6-2
temperature range	operation: 5°C ... 40°C storage: -20°C ... 60°C

Scope of Delivery

- *CAN-Bus Tester*
- CAN adapter cable SUB-D
- cable Open Style
- cable M12
- USB cable, 3 m BNC-cable (1,5m)
- adapter circuit board, for easy connection of oscilloscope probes
- AC/DC adapter with wide-range input
- power cable, 1.8 m
- manual, including CD
- service case

Ordering Information

0800/02 CAN Bus Tester II, CBT II
When ordering please specify the country where it should be used.

CANwatch

Highlights

- fast recognition of installation errors
- preventive diagnosis allows increase of system availability
- scanning of the analog signal and evaluation of the signal quality
- suitable for system integrators and plant operators

Description

CANwatch is an analyzer supporting easy error detection during installation and operation of CAN networks. *CANwatch* judges the analog signal on the bus and detects errors like e.g. Invalid levels, overshoots, slow slopes and short circuits within the signal lines.

Besides the diagnosis of problems during installation *CANwatch* offers the possibility to detect signal deficiencies below the level causing failures. This allows preventive error recognition and thus an increase of the availability of automated systems.

Errors on the physical level of the CAN bus cannot be detected by protocol analyzers and could until now only be analyzed by specialists with expensive and bulky hardware. Due to the easy diagnosis and display with LED's *CANwatch* offers fast detection and correction of errors also to installers and plant operators.

Display

The result of the measurement done at the bus is displayed using 12 different LEDs.

- **CAN OK**
  CANwatch sets this LED with each recognized telegram and resets it cyclically.
- **Serious CAN Failure**
  A basic errors on the CAN is indicated (e.g. short circuit of GND and CAN High or CAN Low respectively).
- **CAN-High locked**
  CANwatch recognizes a short circuit of the CAN High line to a fixed level.
- **CAN-Low locked**
  CANwatch recognizes a short circuit of the CAN Low line to a fixed level.
- **Over-/Undershoots**
  The CAN difference signal of a registered telegram is affected by over-/undershoots in a severe degree.
- **Slow Slopes**
  The CAN difference signal of a registered telegrams is affected by slow signal slope.
- **Invalid Dominant Level**
  A telegram with dominant level above or below the threshold level of ISO 11898 was registered.
- **Invalid Recessive Level**
  A telegram with recessive level above or below the permitted level according to ISO 11898 was registered.
- **Error-Frame Display**
  The indication of the error frames is the percentage of errors compared to the number of registered telegrams. The display is divided into the ranges 0.1 - 1%, 1 - 5%, 5 - 100%.

Technical Data

Limiting Values

Parameter	Min.	Max.
Storage temperature	-20°C	+80°C
Operating temperature	0	+60°C
Supply voltage	-100V	+35V
Voltage on bus connections (Pin 2,7)	-2V	+7V
Admissible power consumption (at 60°C)	-	2000mW

Any (also temporary) stress in excess of the limiting values may cause permanent damage on *CANwatch* and other connected devices. Exposure to limiting conditions for extended periods may affect the reliability and shorten the life cycle of the device.

Nominal Values

Parameter	Min.	Typ.	Max.
Supply voltage	10V	24V	30V
Current consumption	-	40mA	-

All values, unless otherwise specified, refer to a supply voltage of 24V and an environmental temperature of 20°C.

Scope of Delivery

- *CANwatch*
- Manual

Ordering Information

0660/01 CANwatch

CAN Interface Boards

AT-CAN-MINI

Highlights

- plain Basic-CAN/PeliCAN interface
- galvanically isolated from the PC
- connectors compatible with ISO/DIS 11898 and CiA recommendations

Description

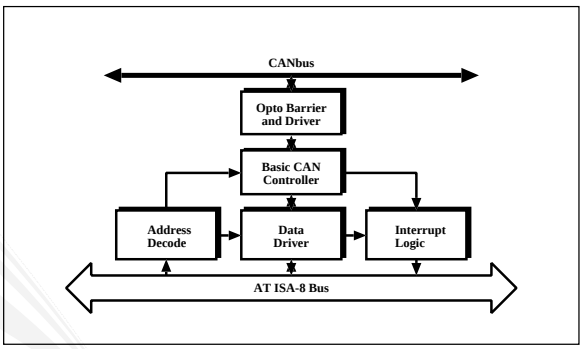
The component *AT-CAN-MINI* is a plain Basic-CAN/PeliCAN interface for PC-AT bus compatible systems.

The component is suitable for all kind of applications, where the main processor provides sufficient computation power to serve the Basic-CAN/PeliCAN interface in an application specific manner.

The PC-CAN interface is realised as a short PC card for ISA bus.

The CAN bus interface is galvanically isolated from the PC.

The card with interrupt ability uses 32 I/O addresses of the PC. The starting address and also the interrupt line are configurable by jumpers.



The interface to the CAN bus is provided by connectors compatible with ISO/DIS 11898 and CiA recommendations.

Technical Data

Power supply	5,0 V, approx. 250 mA
Base address	configurable
Interrupts	configurable
CAN-Controller	SJA1000
Temperature range	0 - 50 °C
Humidity	max. 90%, non-condensing
Size	100 mm x 80 mm
Weight	approx. 100 g
PC bus interface	ISA-8
CAN connector	D-SUB-9m and D-SUB-9f
	CiA-DIS 102

Scope of Delivery

- board AT-CAN-MINI
- manual

Ordering Information

0530/01 AT-CAN-MINI
0530/13 can4linux Device Driver

EtherCAN Gateway

Overview

Connecting "embedded" controllers via the Internet becomes more and more popular in the last time. World wide computer networks are using the common Ethernet cabling standard and TCP/IP as common protocol family.

The reason is the increasing data communication for recording the production and operating data as well as for visualization and remote maintenance.

The *EtherCAN* Gateway combines the decentral CAN and CANopen networks with the classical Ethernet network. To communicate via Ethernet the TCP/IP protocol is used, while on the CAN bus the communication is realized by CANopen, DeviceNet or another user-specific protocol.

Application

Merging the data of production control and data acquisition with operating data lately became a standard requirement of application users.

It is foreseeable, that the real-time control of machines still will be implemented directly with event controlled real time capable field buses for a longer period of time.

Therefore linking the control network over a gateway to office or automation networks or to the Internet at all via TCP/IP is an interesting and cost-effective solution.

This task is fulfilled by the CAN-TCP/IP Gateway *EtherCAN*. The basis for the *EtherCAN* forms a powerful 32-bit CPU with integrated Ethernet controller and the CAN-Controller Philips SJA1000.

Description

On the *EtherCAN* an Embedded LINUX™ operating system is installed. Therefore a lot of services like FTP, Telnet or an integrated web server but also a RS-232 interface are available to run user-specific applications.

For own developments a Developer Package is available. It includes besides tools to generate the FLASH image, a cross compiler and necessary libraries. Further examples in C, Java or Tcl can be downloaded from our web site `http://www.port.de`.

The usage of the *CANopen Library* is possible to create CANopen applications.

Of course ready-to-run programs are available.
For the usage as gateway to CANopen based systems the CANopen Server *m4d* running on the *EtherCAN* realizes the complete handling of the CANopen protocol. It runs as a Network Management Master, can configure LSS slaves with the LSS commands and supports the following CANopen services:

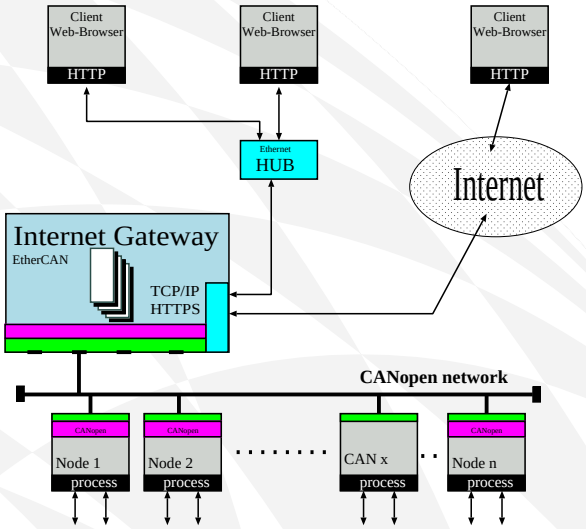
- NMT Services
- Heartbeat, Node Guarding
- SDO Services including Domain Transfers
- PDO Consumer and Producer
- SYNC Producer
- Emergency Consumer

Cyclic services like SYNC of Heartbeat are handled autonomously.

As protocol between a client application and the CANopen Server the CiA specification DSP309-3 is used at the Ethernet side.

For a comfortable handling of the CANopen Server the platform independent and graphical application *CANopen Device Monitor* can be used.

For CAN based systems without High-Layer protocol the *EtherCAN* provides the CAN server *horch* that can send all received data via a TCP/IP port to other applications.



Sending CAN message via the CAN server is possible, too. In the easiest case the server is used as a CAN analyzer via the Ethernet. The CAN Analyzer CAN-RE***port*** can be used for this.

Advantages

The main advantage of such a network linking is the possibility to remote control or do a remote observation and/or service of machines and installations from any place in the world.

The following advantages are to be mentioned further:

- Combination of the advantages from the CAN/CANopen fieldbus technology with the Ethernet
- high transmission capacity in the Ethernet area
- high security and reliability
- no limitation referable the network topology
- TCP/IP is an established standard in information technology
- free access is available to all modules about the network
- low costs per module and for the overall system
- usage of the TCP/IP-protocol

By using the *EtherCAN* gateway also costs for expensive special wirings can be saved, because existing CANopen networks or Ethernet connections can be used again.

Technical Data

Power supply	12 - 30 V
current consumption	max. 150mA
galvanical separation	Ethernet/CAN
microprocessor	32-bit-RISC-ARM 16 MB RAM 2 MB FLASH 80-MHz clock frequency RS232 Interface CAN High Speed Interface
CAN-Controller	NXP SJA1000
CAN	D-SUB-9f D-SUB-9m and D-SUB-9f CiA-DS 102
Ethernet 10/100BaseT	RJ45
serial port	D-SUB-9m
Temperature range	0 - 85 °C
Humidity	max. 90% non-condensing
Size	100mm x 90 mm x 40mm
Weight	approx. 200 g

Scope of Delivery

- *EtherCAN* Gateway
- manual

Ordering Information

0540/07	EtherCAN-CI-ARM7W (Winbond)
0540/08	EtherCANopen-CI-ARM7W (Winbond)
0540/21	Developer Package DEV-PACK-ETH

Functional demo versions of the CAN-RE***port*** and *CANopen Device Monitor* SW tools are available for download on: http://www.canopen-tools.com

IGW/900 Gateway

Highlights

- equipped with an efficient 32-bit Motorola Coldfire with 66MHz clock frequency
- 12-24 VDC Power supply
- CAN 2.0B ISO 11898
- pre-installed Embedded Linux
- full-fledged Linux-Computer on site

Description

With the gateway IGW/900 an equipment is available, which is suitable particularly for the installation in switchgear cabinets. Besides from the compact dimensions IGW/900 is particularly qualified for many industrial applications because of its various connection types. This gateway is equipped with a Linux Device Server. Consequently it can be ascertained a fully-fledged Linux computer is available on site. IGW/900 stays abreast of changes increasing trend to path on high-duty devices to the on site section.

Besides control tasks IGW/900 has the ability to connect devices without Ethernet interface to TCP/IP networks.

Application

The heart of the IGW/900, the DIL/NetPC DNP/5280 also provides with its 32-bit Motorola Coldfire processor with a clock frequency of 66 MHz sufficient power reserves for more extensive applications.

With the combination of IGW/900 as CAN interface and e.g. the CAN analyzer CAN-REport an efficient and versatile tool for analysis and commissioning of CAN-based networks is available. The connection between these two components occurs on the basis of the Client/Server model by standard TCP/IP interfaces. This separation permits to use the entire application over TCP/IP networks away. Therewith the employment becomes possible for remote maintenance without additional or modified software over switched lines or the Internet. This connection between the host system at the Ethernet and the local CAN field bus enabled a comfortable transmission of the technical parameters and facilitate both starting and maintenance.
The Starterkit contains besides the CAN-REport and the CAN Server a development environment to create applications.

Technical Data

Microcontroller	32-bit Motorola Coldfire
Clock frequency	66 Mhz
Memory	16 MB SDRAM 8 MB FLASH
Power supply	12..24VDC
Housing	plastic
Weight	200g
Size	150 x 68 x 25 mm
interfaces	CAN, RS232, RS232/422/485
mounting	top-hat rail/DIN rail

Scope of Delivery

- IGW/900 hardware
- Manual

Ordering Information

0850/01	IGW/900-CAN (single TCP client)
0850/02	IGW/900-CANMC (multi TCP client)
0850/03	IGW/900-CANopen

USB/CAN-Interface CPC-USB

Highlights

- powerful 32-bit microcontroller with internal CAN controller
- supports CAN protocols 2.0A and 2.0B
- display of USB communication and CAN data transfer via LEDs
- supply completely over USB
- galvanic isolation

Description

CPC-USB is an active CAN interface for the connection to the USB bus of a PC. It comes with a robust compact metal housing. The low-price and ease to use makes it well suited for configuration and analysis of CAN networked systems.

The CAN interface supports 11 and 29 bit CAN identifiers. The on-board 32-bit ARM micro controller system Philips LPC2119, with 128 KByte FLASH and 16 KByte RAM, is capable to control transmission and reception tasks with the highest efficiency possible. *CPC-USB* is powered by the USB bus, no additional power supply is required. Development kits for developing applications using *CPC-USB* are available for the operating systems Windows™ and LINUX™.

The CAN connection is galvanically isolated from the USB part. A version with a low-speed transceiver interface for CAN is available to be used in automotive applications.

CPC-USB is useable as CAN interface for the CAN analyser CAN-RE***port***, the configuration tool *CANopen Device Monitor* and the *CANopen Configuration Manager*.

Technical Data

Nominal Values

Parameter	Min.	Typ.	Max.
Current consumption (operating)	-	-	500mA
Supply voltage	4,2V	5,0V	5,5V

Limiting Values

Parameter	Min.	Max.
Storage temperature	-20°C	+80°C
Operating temperature	0°C	+70°C
Supply voltage	0V	+6V
Voltage on bus connections	-30V	+30V

Size: 54 x 24 x 90 mm

Scope of Delivery

- CAN interface CPC-USB
- manual
- CPC-drivers for Windows™ 2000,XP; LINUX™

Ordering Information

0667/01	CPC-USB/M16C-GTI Interface
0667/02	CPC-USB/ARM7-GTI Interface
0690/10	CPC-Series Development Kit/Windows™
0690/20	CPC-Series Development Kit/LINUX™

Functional demo versions of the software tools are available for download on *www.canopen-tools.com*.

USB/CAN-Interface BASIC+ XS

Highlights

- equipped with an efficient microprocessor C165
- the preprocessing of data is possible onboard
- Supports CAN 2.0B ISO 11898-2 High Speed CAN up to 1 MBit/s (82C251) and ISO 11898-3 (ISO 11519-2 Low Speed) Fault Tolerant CAN (TJA1054)
- K-Line ISO 9141 (100 mA) K-Line ISO 14230 (KWP2000)
- drivers as DLL (LevelX) are available SAE J2534 & J2534-1 APIs available
- CAN transfer rates up to 1 Mbit/s

Description

BASIC+ XS is one of the latest addition to the CAN interface technology.
The *BASIC+ XS* interface is a CAN interface for the USB-port or the serial RS232 interface.

An easy "plug in" via USB connection is possible for using this CAN to USB interface.

With "power on" (8...24 V) the microcontroller starts and from FLASH EPROM the USB loader will booted. The *BASIC+ XS* is immediately "ready-to-operate" displayed by a green LED.

The used microprocessor C165 enables the preprocessing of data. It cannot only be used for industrial CAN applications like CANopen, but also for ECU's or vehicle field bus systems. CAN 2.0B ISO 11898-2 for CAN transfer rates up to 1 Mbit/s, ISO 11898-3 Fault Tolerant CAN and K-Line are supported.

LevelX DLL for easy programming of own complex applications under WINDOWS 9x/2000/NT/XP/ME

An easy to use and install CAN monitor application, the PcCANControl is containd in the scope of delivery. High demands are satisfied with the CAN analyser application CAN-RE*port*.

Technical Data

PC bus interface	USB, version 1.1, full speed mode
Microcontroller	C165 @ 18.634MHz
Memory	512k (16bit) RAM 512 byte EEPROM 128k (8bit) FLASH (512k optional)
Power supply	8..24VDC
Power consump.	typ. <100mA @ 12VDC
Temp. range	0 to +55°C
Housing	plastic
Weight	200g
Size	150 x 68 x 25 mm
CAN connector	15-pin Sub-Min-D female

Scope of Delivery

- BASIC+ XS hardware
- RS232 and USB cable (75cm)
- LevelX manual (on CD)
- LevelX driver for Win 2000, XP, Vista

Power supply is not included.

Ordering Information

0703/01	BASIC+ XS Set
0580/10	CAN-REport-W
0580/20	CAN-REport-L

Functional demo versions of the software tools are available for download on *www.canopen-tools.com*.

CAN-Interface CPC-PCI/PCIe

Highlights

- passive CAN interface for PCI- and for PCIe slots
- one or two CAN controller NXP SJA1000
- optional galvanic separation of the CAN channels
- optional separate power supply of the CAN channels with integrated DC/DC converter
- supports 11 bit frames and 29 bit frames
- development kit for Windows™ and LINUX™ available

Description

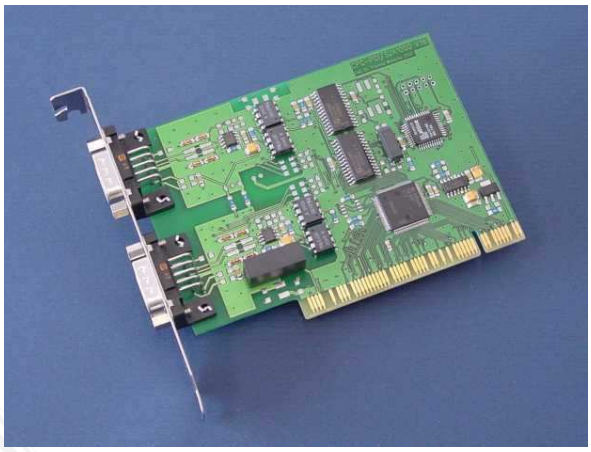
The CAN interface boards CPC-PCI and CPC-PCIexpress were designed for industrial series and has a robust and cost efficient construction. CPC-PCI/PCIe supports either one or two CAN channels that can be operated independantly with different data rates. The interface comes with the NXP CAN controller chip SJA1000, that offers good diagnostic attributes.
CPC-PCI/PCIe maps the CAN controller(s) directly in the address space of the PC and allows access of the CAN messages with low latencies. Existing software for the supported CAN controller can easily be adapted. The CAN communication both with CPC-PCI and with CPC-PCIe may be handled either in interrupt controlled mode or in polled mode, interrupt channels are assigned automatically.

Optionally, both CAN interfaces CPC-PCI and CPC-PCIe are available with galvanic separation between PC and CAN bus. A galvanic separation between the CAN channels is also possible by separate DC/DC converters.

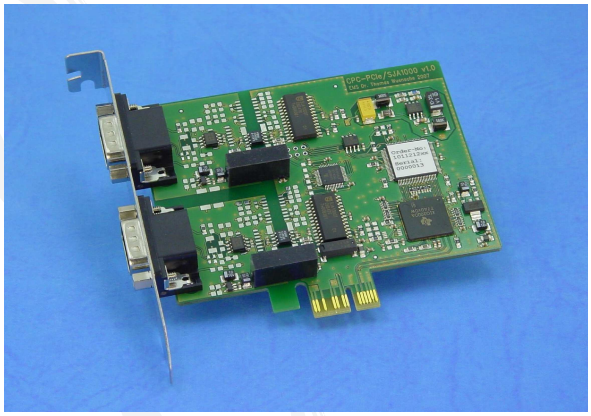
Technical Data

Bus Interface

Pin assignment	Connector DSub 9 complying to CiA DS-102
Type of the phy. connection	ISO 11898 Transceiver PCA82C251
Maximum voltage on bus pins	±30V refering to bus ground
Isolation voltage with galvanic separation	±1000V



picture 1 CPC-PCI Interface



picture 2 CPC-PCIexpress Interface

Programming Interface

Configuration and CAN communication with CPC-PCI and with CPC-PCIe are done by accesses to the memory address area of the PC. The appropriate data areas are mapped by CPC-PCI/PCIe and "plug & play" software into the memory address space.
The memory used by CPC-PCI/PCIe is divided in several sections. The control registers allow the detection of the interface type similar to the interfaces CPC-Card, CPC-XT and CPC-104 and the generation of interrupts for each CAN controller separately. The access to the CAN controllers occurs via a separate memory space for each interface. In these memory spaces, the sending and receiving buffers as well as the control registers of the CAN controllers are accessible.

Scope of Delivery

- Plug in board CPC-PCI/PCIe
- manual
- Sample code
- CPC-drivers for Windows™2000,XP,Vista
  LINUX™

Ordering Information

0665/01	CPC-PCI/SJA1000S-GTIS 1-channel
0665/02	CPC-PCI/SJA1000D-GTID 2-channels
0669/01	CPC-PCIe/SJA1000S-GTIS 1-channel
0669/02	CPC-PCIe/SJA1000D-GTID 2-channels
0690/10	CPC-Serie Development Kit/Windows™
0690/20	CPC-Serie Development Kit/LINUX™

PCI/CAN-Interface PCI-IntelliCAN

Highlights

- intelligent PC interface board with fast 16-bit microcontroller for real-time applications
- support of 2 channels CAN Part 2.0B active
- easy interfacing between PCs and CAN-networks
- Hardware configuration via PCI software services, plug-and-play
- data monitoring
- on-board data (pre-) processing

Description

The increasing number of PCI slots and the high demand on real-time networks dominate today's PC world. The dual-channel *PCI-IntelliCAN* board is a special design for PCs with PCI slots. The *PCI-IntelliCAN* is designed for real-time data acquisition and processing by using a fast 16-bit microcontroller

Hardware configuration via PCI software services (plug-and-play). A DPRAM interface for host interfacing is available. The customer can use LevelX (DLL) for the easy programming of own complex applications. With the *PCI-IntelliCAN* a powerful hardware solution is available for a CAN interface card.

Technical Data

size	108 x 155 mm
power supply	from PC
temperature range	0 to +70°C
processor types	C165 @ 20 MHz
memory	8 kB DPRAM 16 bit 12 kB SRAM 16 bit 28 kB FLASHROM 8 bit optional: 512 KB
CAN controller	SJA 1000
operating system	Windows9x,NT,2000,XP
CAN connector	CAN I: 9-pin Sub-Min-D, CiA standard CAN II: 9-pin Sub-Min-D, CiA standard
PC bus interface	PCI-BUS
PIN-Connector	- serial sync. - serial async. - 4 x digital in/out - 5 x digital in - optional 8x analog in (10 bit resolution; 0 V to 5 V)

Scope of Delivery

- board PCI-IntelliCAN
- user manual (on CD)
- LevelX user manual (on CD)
- LevelX drivers for Windows™2000, XP, Vista
- examples for Windows

Ordering Information

0701/01	PCI-IntelliCAN

can4linux - Linux Device Driver

Overview

can4linux is an universal LINUX™ device driver. It can be used to control ISA or PCI interface boards with CAN interface, or embedded CAN controller solutions. The GPLed sources are currently supporting CAN controllers (Philips SJA1000, Intel 82527, Infineon SAK82C900 and Motorola FlexCAN, TwinCAN). The driver can be compiled for all LINUX™ kernel versions 2.4 to 2.6. (older version for kernel versions 2.0 to 2.2 are available.)

Application

The *can4linux* Project started already in 1997 in cooperation with the LINUX™ LLP project to control laboratory or automation devices via CAN. Derived from this driver is the new *can4linux* version line 2.x and follow-up. Designed to be used in embedded applications as well as on the desktop this version is configured at compile time for the right hardware. Originally with support only for SJA1000 there is now support for Intels 82527 and new support for Infineons TwinCAN. Within the scope of the uClinux project a version supporting Motorolas FlexCAN was created.

Driver Interface

Using the LINUX™ kernels `/proc` interface, the driver is highly configurable. The driver is controlled by the standard driver functions `open()`, `close()`, `read()`, `write()` and `ioctl()`.

Any number of CAN interfaces can be accessed using the device nodes `/dev/can0` ... `/dev/can`*n*. *can4linux* uses the device `major` number 91.

Hardware

The driver supports different interface boards, ISA, PCI or PC-104, with one or two channels.

- ***port*** AT-CAN-MINI (PeliCAN mode 29bit)
- EMS CPC-PCI
- IXXAT PCI03 (only SJA1000)
- Advantech Pcm-3680 (PC 104)
- DIL-NET TRM/816
- DigiTec PCMatic® Feldbusmodul FC-CAN
- ESD PC104/200
- Blackfin BF534/536

Originally the driver worked only on Intel-based PCs. In the mean time we have gained experiences with porting this driver also to embedded LINUX™ architectures. One is a PowerPC 863e with two Intel 82527 controllers, the other is ARM based with an external SJA1000, the microcontroller XC164CS with TwinCAN controller or the NetARM 150 with its four internal CAN channels.

Available Products

can4linux can be used as CAN interface for different products of ***port***. Some of these tools are available as demo versions, to be downloaded from our web site. ⟨`http://www.port.de/shop.html`⟩

The simplest application with access to the CAN layer-2, that means transmit and receive CAN messages, is *horch*. It can be started interactively within an *xterm* or in the background as TCP/IP Server. For this server a graphical and programmable CAN-analyzer application CAN-RE***port*** is available too.

To implement CANopen protocols for devices in CANopen networks, an ANSI-C communication library is available for development of own applications. Based on the library code, a so-called CANopen Server application is available. The Server provides easy access to CANopen services through a TCP/IP socket interface.

System Requirements

can4linux is freely available under the GPL license. It can be compiled for the LINUX™ kernels 2.0 to 2.6 for different hardware architectures.

Ordering Information

0530/13 *can4linux* Device Driver

can4vxworks - Device Driver for VxWorks

Overview

The *can4vxworks* driver is a universal VxWorks device driver and is used for controlling of ISA-, PCI- and PC104 interface cards with CAN bus interface. It supports the CAN controller Philips SJA1000 and the sources can easily be adapted to use any other CAN controller .

Application

The *can4vxworks* driver is a result of the *can4linux* project which was developed in 1997 to control laboratory devices that use CAN. Besides the IOCTL function calls both drivers use the same data structures for parameter passing. Thus changing an application between LINUX™ and VxWorks is trouble-free.

During compilation the driver is optimized for the target hardware.

Driver Interface

Controlling the device driver is done via the standard function calls for devices like `open()`, `close()`, `read()`, `write()` and `ioctl()`.
The driver is highly configurable with the IOTCL functions.

Furthermore it is possible to use up to four interface boards with the device files /dev/can0 ... /dev/can3.

Hardware

The driver supports various single and dual-channel standard PC interface cards with ISA, PCI and PC104 interface:

- ***port*** AT-CAN-MINI (PeliCAN mode 29bit)
- EMS CPC-PCI
- IXXAT PCI03 (only SJA1000)
- Advantech Pcm-3680 (PC 104)
- DigiTec PCMatic® Feldbusmodul FC-CAN (SJA1000)
- ESD PC104/200

The driver was developed for x86 machines but can also be used on other target hardware.

Products available

The *can4vxworks* driver can be used with different products of ***port***.

In order to send, receive and display CAN messages the command line application *horch* is available as CAN layer 2 server. It can be used interactively from within an xterm or in batch mode as TCP/IP server. In the latter mode the graphical CAN analyzer CAN-RE***port*** can be used to send, receive and display the CAN messages.

Based on this source code the compiled and configured CANopen server *m4d* is available. It also provides an interactive and a batch server mode. To use it in server mode ***port*** offers the tool *CANopen Device Monitor* with an easy to use graphical interface.

System

The *can4vxworks* device driver is available as ANSI-C source code and can be compiled on different target platforms.

Ordering Information

0643/01 can4vxworks Device Driver

Software Components for Ethernet POWERLINK

Introduction to Ethernet POWERLINK

Ethernet POWERLINK, originally developed by the Austrian specialist Bernecke&Rainer (B&R), is advanced by leading automation companies under the management of the EPSG - the Ethernet POWERLINK Standardization Group. It is one of relevant standards in Europe for the use of Ethernet in the automation industry. Through its past history Powerlink is already very common. More than 100,000 nodes have worked without interference in series machines and plants for several years worldwide. **Ethernet POWERLINK**, is a communication profile for extension of the IEEE 802.3 (Fast Ethernet) for automation. The EPSG standard is already brought in the IEC and accepted to the standardization.

ETHERNET POWERLINK Standardization Group

The great attraction of Ethernet is the possibility of taking advantage of the medium from the sensor level up to the control center level. Powerlink greatly values it, that however the user can concentrate on his application and isn't blocked by network administration. An automation technician still can exchange his components with the screwdriver. The address allocation is solved simply, but nevertheless the real time devices can be reached corporation-wide by there IP-addresses.

Targets of the EPSG aren't only high speed connected with real time ability and IP integration.
On the application layer a high openness is advanced regarding the device profiles. A cooperation exists with the CiA - CAN in Automation - to integrate the CANopen communication and device profiles in **Ethernet POWERLINK** together. The combination of the advantages of real time Ethernet using POWERLINK and the CANopen profiles offers a continuity from the simple sensor network, over fast motion control systems up to the factory network. The integration of the open device profiles is already completed. Service providers like ***port*** GmbH offer software tools and services for the POWERLINK-device development. Meanwhile the EPSG works on useful extensions like the time synchronization in complete enterprise networks and the integration of security relevant functions.

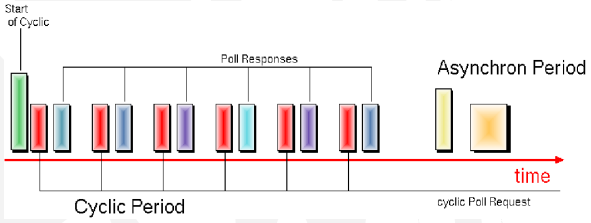
POWERLINK development goals

- Fast Ethernet according IEEE802.3u 100BASE-TX as a transmision medium.
- Use of standard network-hubs and standard cables.
- deterministic transmitting of cyclical data with a minimal cycle time of 200 µs
- Jitter smaller than 1 µs
- Transmitting of deterministic and time uncritical data
- Use of standard IP protocols (TCP, UDP, HTTP)

POWERLINK mode of operation

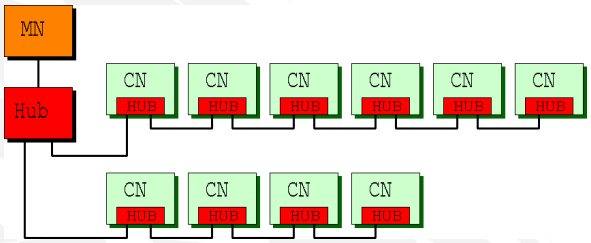
In POWERLINK all data transmissions are checked by a manager «Managing Node» MN to avoid frame collisions in the Ethernet network.

The network devices, «Controlled Node» CN, only send when they are asked for it by the manager. An POWERLINK cycle is divided into two time domains. A «Start of Cyclic» (SoC) frame, sent by the manager as a broadcast message to all controllers, starts the strictly deterministic «Cycle period». In this cycle an isochronous data interchange is carried out. The Managing-Node sends one «Poll Request» in the Unicast mode sequentially to every Node.



This is answered as fast as possible with a «Poll Response» by the addressed node. The answer is sent as a Ethernet Multicast frame and therefore cannot only be received by the manager but by every Node having been interested in it. The «Asnychronous Period» follows directly. The Managing Node asks here again every Controlled Node one by one with a Ethernet unicast «Invite» frame to send data. E.g. the nodes can send IP frames in this part of the cycle. Arbitrarily many topologies are practicable by the use

of Ethernet-Hubs. From external Legacy Ethernet and other IP networks one can access directly POWERLINK devices via special POWERLINK-capable routers.



The introduction of an CANopen compatible Application Interface in plants and machines also makes the simple migration of CANopen to Powerlink or the combination of both busses possible.

POWERLINK V2 Properties

- up to 240 devices in one POWERLINK network line
- POWERLINK Manager: 240 TX-PDO, 240 RX-PDO
- PDO Mapping with up to 254 objects
- maximum size of a PDO is 1490 bytes
- Every device can communicate with every other one via SDO
- Error messages with detailed error information
- Standard Boot-Up behavior and configuration manager are defined

POWERLINK Software

The standardization of the version 2.0 is now completed and the standard part of the IEC documents (IEC 61784.) Many companies have already running implementations. On fairs Multivendor plants are shown with promising results. ***port*** GmbH provides software for the implementation of Controlled Nodes according V2.0 with CANopen application interface already now. There are already example implementations available of an POWERLINK-Controlled Node for the Freescale processor MC9S12NE64, Xilinx FPGAs and for the STR910 of ST Microelectronic.

The port POWERLINK-protocol stack is available in a generic version which simply can be adapted to different target systems and operating systems.

The current status of the standardization of **Ethernet POWERLINK** and available knowledge and tools requires a close cooperation between device designers and Stack providers at the present time. ***port*** offers its services and experiences to this to all it's customers.

Ethernet POWERLINK Library

Overview

The POWERLINK *Library* provides all required services for a compliant communication according to the POWERLINK communication profile V2.0. It facilitates easy and fast development of Controlled Node and Managing Node devices.
The POWERLINK *Library* is available in different expansion stages:

- Controlled Node
- Managing Node

The functional range can be extended by additional modules (Redundancy, POWERLINK safety, ...) (see *POWERLINK-Library Extension modules*). These modules are currently in a planning phase. Access to the hardware is carried out via a hardware abstraction layer. By the nature of POWERLINK it is necessary to obtain very short response times. Therefore these drivers are highly optimized and currently only available for selected Ethernet MAC controllers.

Application

The POWERLINK *Library* is based on ground of the POWERLINK V2.0 communication profile of the ETHERNET Powerlink Standardization Group EPSG. It provides the most importand services specified therein.

It is completely written in ANSI-C and can be compiled with every ANSI-C compliant compiler.

Depending on the required scope of functionality the POWERLINK *Library* is available in different expansion stages. With the standard *Controlled Node* version of the POWERLINK *Library* all services are provided for development of full-featured CN devices.
The functionality of the *Managing Node* as well as the comfortable node monitoring functionality is provided by a different product.

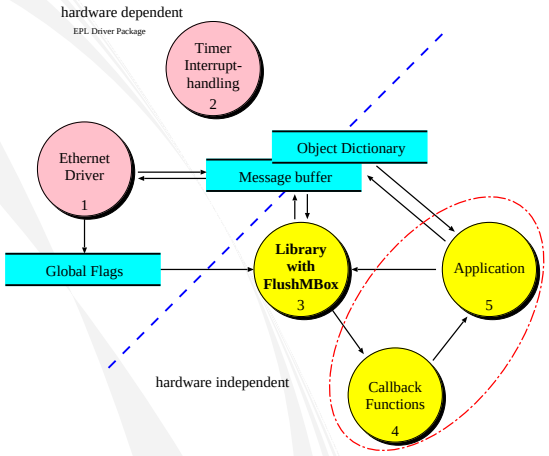
Additional services of other communication profiles are provided by means of extension modules.

All hardware specific parts are decoupled from the POWERLINK protocol stack through a defined driver interface. This provides easy adaptation to different hardware platforms.

The user application communicates with the POWERLINK *Library* through function calls and call-back functions.

Configuration and scaling of the POWERLINK *Library* is done with the help of the POWERLINK *Design Tool*, which is delivered as POWERLINK *Design Tool Light* version. With it the POWERLINK *Library* can be tailored to an optimum to the available resources of the application. Besides the creation of the object directory all settings for the hardware can be carried out with it.

Description



The POWERLINK *Library* consists of a hardware independent and a hardware dependent part that communicate with message queues. The hardware dependent part consists of controlling software for the Ethernet MAC controller and timer functions.

The application communicates only with the hardware independent part of the POWERLINK *Library*. That way it is possible to exchange drivers without any influence on the functionality of the application. The initialization of POWERLINK services is done with function calls within the application. During the execution of the application the POWERLINK *Library* executes all necessary communication tasks autonomously and informs the application about successful received POWERLINK messages with the help of callback functions.
Communication requests from other devices as well as necessary periodical tasks and time out monitoring is handled within the POWERLINK *Library*. All requests are proved for correctness (access rights, data types etc.). The application is notified after completion of the communication and occurrence of failures, respectively, through service oriented callback functions. In these callback functions appropriate actions can be carried out from the application.

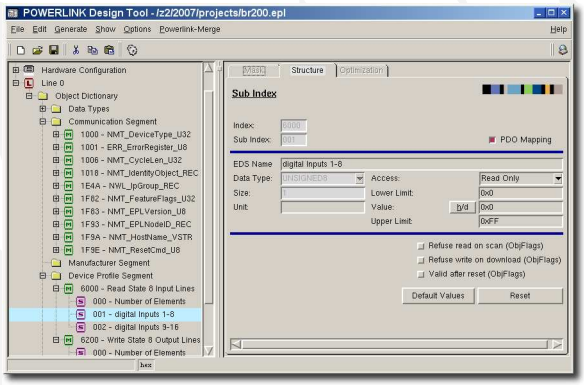
The object directory is designed to contain references to the variables in the user application. Consequently it is possible to take over variables from an existing software without any changes in the object dictionary. The high degree of scalability of the POWERLINK *Library* is of particular importance for devices with limited resources. On one hand, this is achieved by the modularity in individual service groups, and on the other hand, through the use of compiler directives in the respective modules. Thus, the code size is proportional to the utilized POWERLINK services.

Delivery of the POWERLINK *Library* comprises different example programs that describe the usage of the various POWERLINK services. All examples contain a complete implementation of a POWERLINK device including the object directory as well as application code. These are ready to be compiled and run.

Among the detailed documented source code there is the reference manual and a printed user manual as documentation available.

The POWERLINK *Library* is constantly improved and adopted to customer requirements. In order to keep up with the latest version of the POWERLINK *Library* ***port*** provides its customers an update service.

For development, test and initial operation of POWERLINK devices ***port*** provides a comprehensive tool chain. The creation of the object directory is simplified with the POWERLINK *Design Tool* (POWERLINK *Design Tool Light* version belongs to the scope of delivery).



The graphical POWERLINK *Design Tool* is available for the creation of the object directory, the electronic data sheet (EDS file in XML format) and for the documentation of the device in HTML or text format.

The POWERLINK *Device Monitor* can be utilized for commissioning, but also for the implementation and test phase.

For the development of POWERLINK application monitoring and analyzing the POWERLINK bus traffic is essential. The service oriented display of POWERLINK messages of the POWERLINK-RE***port*** allows easy and quick interpretation of POWERLINK messages.

Besides the communication objects several application objects in different device profiles are also specified in POWERLINK. These definitions guarantee a defined behavior of the corresponding device class and enable the interchangeability of POWERLINK devices. Device profiles can be easily created by use of the POWERLINK *Design Tool*.

Scope of delivery

- POWERLINK library with separate driver interface
- Ethernet MAC driver
- immediately compilable examples
- POWERLINK *Design Tool Light*
- user manual
- reference manual containing descriptions of all functions, including parameters and return values

Further Services

- Maintenance Agreement
  All changes of current standards as well as extensions through new developed standards are constantly incorporated into the POWERLINK *Library*. In order to take profit of the changes ***port*** offers all its customers a maintenance agreement with the following conditions:
  - ☐ updates free of charge for the contracted period
  - ☐ free of charge support for the POWERLINK *Library*
- Support for initial operation
  In order to provide a quick and effective access to the development of POWERLINK devices we recommend to do the initial operation of the POWERLINK device together on the target platform. Customer experiences of his/her hardware, the used compiler and the experience of our engineers with POWERLINK and the POWERLINK *Library* can complement each other. This leads to reduced development times and a POWERLINK conform device.
- Training
  Trainings can be given at ***port*** or at your companies rooms and are exclusively carried out for your developers.

Ordering Information

1700/01	POWERLINK-SRCLIB Controlled Node ANSI-C Source Code
1710/10	POWERLINK-SRCLIB-CN-NE64 CN Stack für Freescale NE64
1710/11	POWERLINK-SRCLIB-CN-STR910 CN Stack für STR910-Evaluation Board
1710/12	POWERLINK-SRCLIB-CN-Xilinx-Microblaze CN Stack for Xilinx-Microblaze with PE²MAC from ***port***

POWERLINK StarterKit

Overview

It is often wished to get to know the POWERLINK protocol, devices and tools without having high expenditure. This wish is met with ***port*** GmbH's POWERLINK StarterKit (PSK).

The StarterKit enables the device developer to explore the capabilities and the advantages of ***port***s POWERLINK Library.

The PSK does not only allow you to get to know the POWERLINK protocol, but it also supports the users when testing POWERLINK devices and networks.

With prepared and modificable examples developers can estimate the effort needed to create own applications.

Description

POWERLINK is a very flexible Real-Time-Communication protocol. Its main advantage is the use of standard Ethernet components for 100Mbit/s Ethernet like PHY, Ethernet-MAC-Controller (EMAC), cables and other infrastructure components for the devices Controlled Nodes (CN, the **slaves**) and Managing Nodes (MN, the **master**).

Using FPGAs opens up new possibilities. Best real time behaviour short delays and short response times are achieved by the optimized Ethernet MAC controller that is supported by the hardware. FPGAs meet the demand for a built in hub in order to connect devices in a line network structure like conventional field busses.

port offers several reference platforms based on standard Ethernet controllers, as well as the FPGA solution. So, beside the evaluation of the POWERLINK protocol, it allows to compare different hardware platforms.

With the hardware of choice it is possible to simulate process elements directly via control elements and give the outputs to LEDs. On side of the POWERLINK

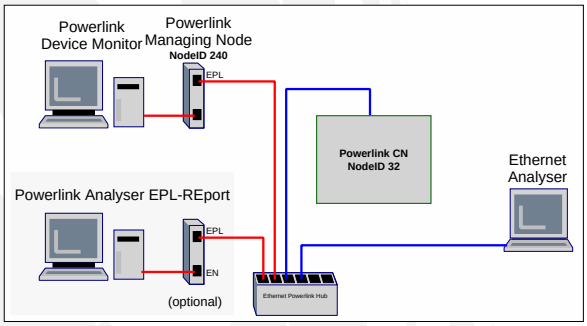
the device behaves like a IO-device standardised to CiA 401, which can be found in the industrial field.

Hardware

The PSK includes:

- target hardware for program development of a CN, XILINX Spartan3E-500 FPGA, Freescale NE64 or ST Microelectronics STR910
  including development environment and examples
- POWERLINK Device Monitor *(demo version)* as POWERLINK Managing Node,
- a Windows based Ethernet analyzer with POWERLINK Plug-In
- POWERLINK Hub and Ethernet cable
- optional the **POWERLINK**-RE***port*** *(demo version)* as powerful POWERLINK analyzer

All software components are available on the enclosed Tools CD.



System Requirements

The usage of the starterkit requires a PC system or Laptop with at least 3 GByte free disc space as well as Microsoft Windows XP Professional or Vista Business

Ordering information

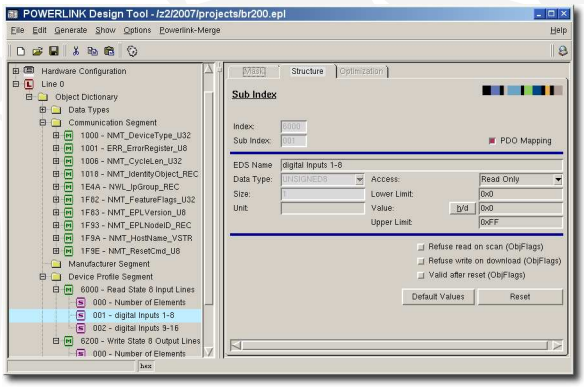
1701/01	PSK-Spartan3E target hardware for program development of a CN, XILINX Spartan3E-500 FPGA including development environment and examples
1701/11	PSK-Spartan3E+ as PSK-Spartan3E including an additional POWERLINK Gateway
1701/02	PSK-NE64 target hardware for program development of a CN, Freescale NE64-Board including development environment and examples
1701/12	PSK-NE64+ as PSK-NE64 including an additional POWERLINK Gateway,
1701/03	PSK-STR910 target hardware for program development of a CN, Microelectronic STR910 including development environment and examples
1701/13	PSK-STR910+ as PSK-STR910 including an additional POWERLINK Gateway

Functional demo versions of the software tools are available for download on our web site *http://www.epl-tools.com/shop.html*.

POWERLINK Design Tool

Overview

The POWERLINK *Design Tool* is a tool for the rapid and cost-saving development of **Ethernet POWERLINK** applications (devices). It administers device databases, from which an object dictionary and an initialization function in C-code, an Electronic Data Sheet (XML format) and the documentation are produced automatically. Furthermore it simplifies the configuration of the POWERLINK *Library* and of the *POWERLINK Driver Packages*.



Description

With the POWERLINK *Design Tool* a tool is available, which frees the developer from error-prone and repetitive tasks. It guarantees the consistency of the implemented functionality, device documentation and Electronic Data Sheet (EDS). By default the POWERLINK communication profile is delivered with the tool.

In addition to the POWERLINK specific communication database other databases are provided for the CANopen device profiles CiA 401, CiA 402, CiA 406 and CiA 417. The object dictionary produced by the POWERLINK *Design Tool* supports numerous options of the POWERLINK *Library* from ***port***, e.g. segmented structuring. A tree representation of all implemented parameters and data facilitates the maintenance of the device software. The POWERLINK *Design Tool* eases the first step into the POWERLINK protocol and device development is accelerated.

It is a tool with which device databases can be produced and processed. These databases contain information, which describe the interface for the POWERLINK network of the device. These parameters, control and status information that are accessible over the network are stored in the **Object dictionary**.

The major task of the POWERLINK *Design Tool* is to administer and create these data in an object database and to create code for the POWERLINK *Library* from it.

Ethernet POWERLINK Object dictionary

The generated code, which contains the object dictionary is included into the user modules. This ensures that each user variable can be accessed directly via its index and sub index. The produced object dictionary represents the data interface between library and user appliction.

Ethernet POWERLINK Library Configuration

To configure the POWERLINK *Library* a header file is created which contains the hardware settings and the configuration of the used POWERLINK services. These services are derived automatically from the object dictionary.

EDS File

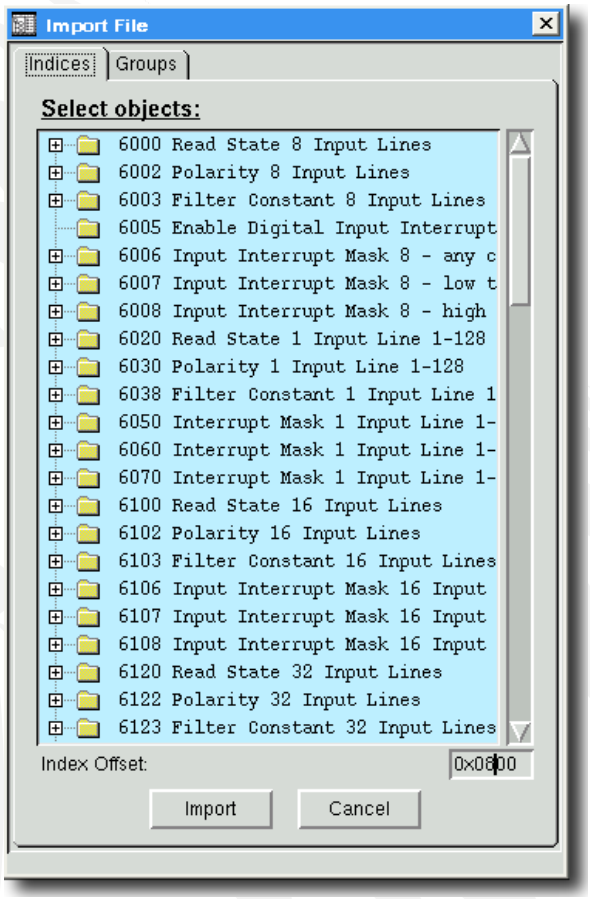
A fourth result of the tool is the Electronic Data Sheet (EDS). The EDS belongs to the standard documents that have to be supplied with a POWERLINK device. It is an XML text file according to ISO 15745, which contains all relevant information about the application (device). These files are used by configuration tools and control devices, in order to install POWERLINK devices into networks and configure them. The

Documentation

On the generated object dictionary a documentation is created. Each device parameter is listed with its attributes and description. Output format is HTML and plain text. This documentation can serve as specification at the beginning of development and be brought into the user documentation, as well as into the marketing documents.

Device Profiles

Creating of device databases is very simple. The first step is determining the type of the device. If a standardized profile is to be implemented, then the device data can be loaded from a profile database. Furthermore it is possible to load data from device databases or already existing EDS files. EDS files can be imported.


Import File
Indices
Groups
Select objects:
6000 Read State 8 Input Lines
6002 Polarity 8 Input Lines
6003 Filter Constant 8 Input Lines
6005 Enable Digital Input Interrupt
6006 Input Interrupt Mask 8 - any c
6007 Input Interrupt Mask 8 - low t
6008 Input Interrupt Mask 8 - high
6020 Read State 1 Input Line 1-128
6030 Polarity 1 Input Line 1-128
6038 Filter Constant 1 Input Line 1
6050 Interrupt Mask 1 Input Line 1-
6060 Interrupt Mask 1 Input Line 1-
6070 Interrupt Mask 1 Input Line 1-
6100 Read State 16 Input Lines
6102 Polarity 16 Input Lines
6103 Filter Constant 16 Input Lines
6106 Input Interrupt Mask 16 Input
6107 Input Interrupt Mask 16 Input
6108 Input Interrupt Mask 16 Input
6120 Read State 32 Input Lines
6122 Polarity 32 Input Lines
6123 Filter Constant 32 Input Lines
Index Offset:
0x0000
Import
Cancel

Import mask for a profile database

The second step is the parameterization of the objects. Pre-installed default values, limit values and access attributes have to be adapted to the application needs. For the communication parameters input masks facilitate the creation of the service objects and the configuration of partially bit-coded values like e.g. PDO settings.

Generation

After an automatic check of the data for consistence, missing entries or incompatible data types the following files are created:

- the object dictionary (**objects.h**),
- the configuration file (**epl_conf.h**),
- the XML EDS file
- and documentation as HTML and plain text.

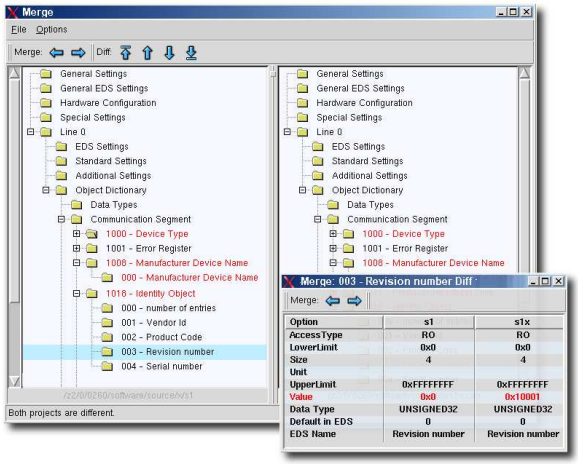
Profile Databases

The profile databases contain the objects of the corresponding profile. Each object is configured according to the standard and is explained in detail. By importing these prefabricated objects into the own project the development time can be reduced and the risks of a manual implementation are minimized.
The following profile databases currently available:

- CiA401 - Device Profile for Generic I/O Devices
- CiA402 - Device Profile for Drives and Motion Control
- CiA404 - Device Profile for Measuring Devices and Closed-Loop Controllers
- CiA417 - Application Profile for Lift Control Systems

POWERLINK-Merge

The optionally available POWERLINK-Merge PlugIn is a tool to compare and to merge POWERLINK *Design Tool* projects. A detailed presentation of each attribute of indices and sub indices and comfortable editing facilities simplify the handling of multiple projects. The integrated diff report generator generates expressive diff reports in XML.



POWERLINK-Merge shows its strength especially when dealing with large projects or with large development teams.

Advantages

- acceleration of the development by avoidance of double work
- error prevention during creation of the object dictionary and EDS
- consistent documentation to the respective device version
- reuse of the data for similar devices
- optimized configuration of the POWERLINK *Library* according to the implemented POWERLINK services

System Environment

The POWERLINK *Design Tool* runs on PC's with

- Operating System: Windows™ 2000, XP, Vista and LINUX™
- Processor: Pentium III or later
- RAM: > 256 MB, HD > 20 MB free

Ordering Information

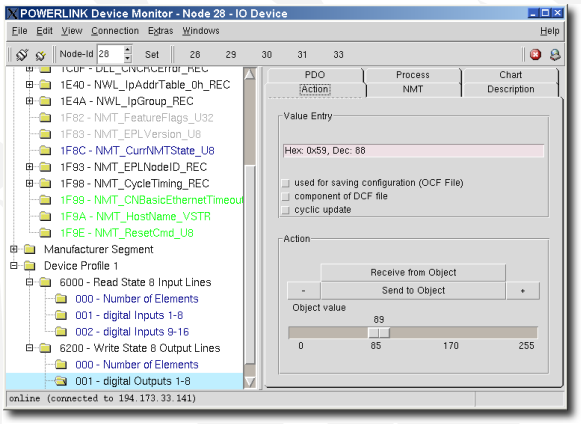
1760/10	POWERLINK-DT-W (Full Version Windows™)
1760/20	POWERLINK-DT-L (Full Version LINUX™)
0640/11	CANopen-DT-P401 profile database for CiA401
0640/12	CANopen-DT-P402 profile database for CiA402
0640/13	CANopen-DT-P406 profile database for CiA406
0640/23	CANopen-DT-P404 profile datebase for CiA404
0640/30	CANopen-DT-P417 profile database for CiA417

Functional demo versions of the software tools are available for download on our web site *http://www.epl-tools.com/shop.html*.

POWERLINK Device Monitor

Overview

The POWERLINK *Device Monitor* is a tool for the graphical inspection and configuration of POWERLINK devices in a POWERLINK network. The embedded scripting ability makes it possible to access the implemented POWERLINK services and to write test or control applications with a minimum of effort.



The information on how to display the device object dictionary is obtained from the EDS file of the device (EDS - Electronic Data Sheet according to ISO 15745) or can be scanned from the device directly.

Description

The program's graphical user interface consists of

- a hierarchical tree representation of the object dictionary
- various action fields
- an optional command console interface

Displaying all implemented objects as a tree facilitates the configuration by hand. The treeview is divided into the following segments:

- communication profile
- manufacturer-specific objects
- device profiles

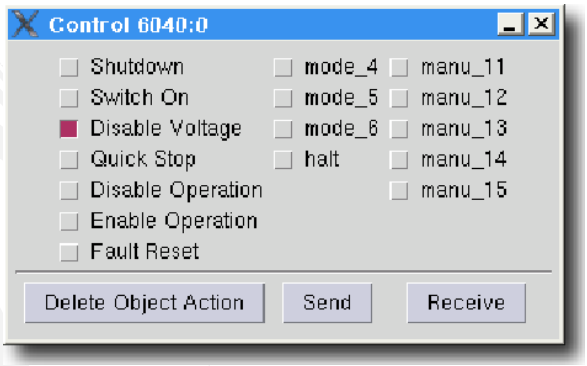
Devices with up to 8 device profiles are supported.

The `tab Action` is used for reading and writing device objects with POWERLINK SDO transfer messages.

Supported standard actions are:

- reading of a device object
- writing of a device object
- incrementing of a device object
- decrementing of a device object

Optionally a slider with a random value range can be assigned to every object to achieve a more useful configuration. For each object the meaning of each bit can be displayed.

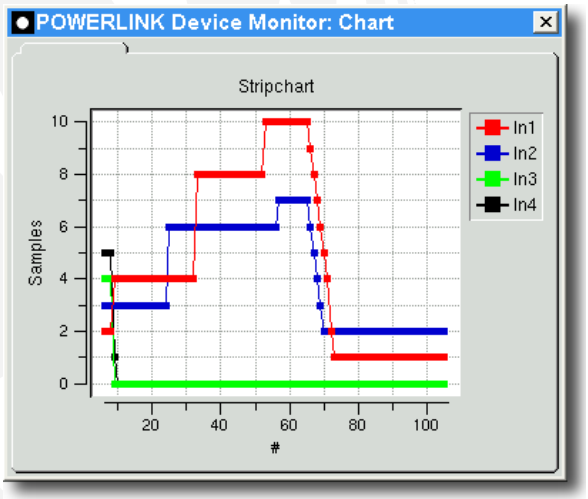


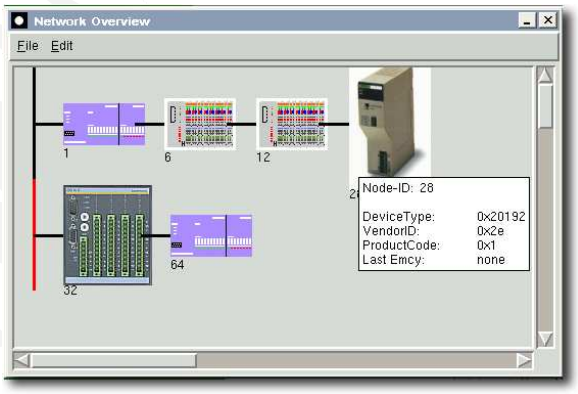
Values of manually configured device objects can be stored to a file and reloaded into the device.

The `NMT tab` makes several fields available for the execution of NMT services.

Under the `tab Description` an object description of the currently selected object in the object tree is shown. The description is taken from a text file which is created by the POWERLINK *Design Tool* or manually written with a common text editor.

`Additional tabs` are available for graphical representations of process data and can be defined by the user.





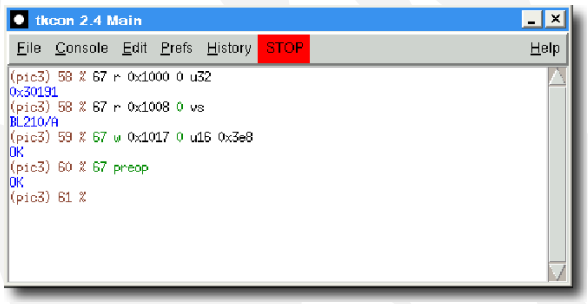
Network Overview

The network overview displays the devices in an POWERLINK network.

The console is an extra feature. With the console an interactive access to the POWERLINK network is possible by using commands for:

- SDO transfers
- PDO transfers
- network management services

Complete sequences of commands can be combined to procedures by means of the embedded scripting language.


tkcon 2.4 Main
File Console Edit Prefs History STOP Help
(pic3) 58 % 67 r 0x1000 0 u32
0x30191
(pic3) 58 % 67 r 0x1008 0 vs
BL210/A
(pic3) 59 % 67 w 0x1017 0 u16 0x3e8
OK
(pic3) 60 % 67 preop
OK
(pic3) 61 %

This feature is particularly useful for POWERLINK device developers. Through the scripting language all implemented POWERLINK communication services are available for test and control applications. Logging of the test sequences is also possible.

By using the POWERLINK *Library* and the POWERLINK *Design Tool* from ***port*** the developer gets a closed tool chain for creating POWERLINK compatible devices.

Through the scripting language access to elements of the graphical interface is available too. The graphical representation of values or test results is therefore application specific extensible.

port GmbH can perform special adaptions to the GUI or behavior to meet POWERLINK device characteristics for service or assembly testing.

POWERLINK-Interface

The POWERLINK *Device Monitor* requires a hardware interface. It's a special POWERLINK/Ethernet router using the synchronous **Ethernet POWERLINK** on one side an on the other side it provides its services via TCP/IP.

System Requirements

Operating System:	Windows™ 2000/XP, Vista LINUX™
Processor:	Pentium III and better
RAM:	256 MByte
Harddisk Space:	>20 MByte available

Ordering Information

1770/10	POWERLINK-DM-W (Full Version, Windows™)
1770/20	POWERLINK-DM-L (Full Version, LINUX™)
0640/50	CANopen-EDS2XML-W EDS-to-XML converter Windows™
0640/51	CANopen-EDS2XML-L EDS-to-XML converter LINUX™

Functional demo versions of the software tools are available for download on our web site *http://www.epl-tools.com/shop.html*.

POWERLINK-REport

Overview

In daily use of POWERLINK it is important to determine very fast and reliable the main parameters of the network to recognize potential possible error sources or really faulty devices or configurations.

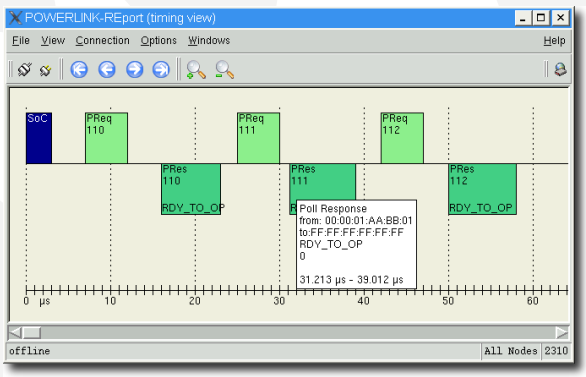
The POWERLINK-RE*port* is a versatile POWERLINK network analyzer and monitor. With its features for logging, analysis and interpretation of POWERLINK messages it supports development, diagnostic and configuration of POWERLINK devices and networks.

	Property/Parameter
1	Cycle time
2	Slot time
3	Response time
5	Time between two slots
5	Length of the individual periods of the cycle
6	Number of isochronous package couples
7	Order of the isochronous enquiries
8	Exploitation of the asynchronous phase
9	Nodes in the net
10	Discarded packages

Description

The timing view displays the messages of a POWERLINK cycle over a time line. The following attributes of a message are shown:

- Type of frame (SoC, PReq, PRes, SoA, ..)
- Source or destination
- NMT status (if applicable)



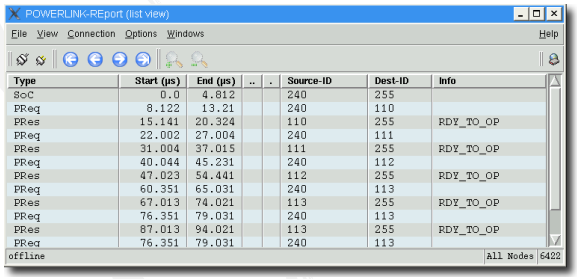
The POWERLINK version V1 or V2 is detected automatically and the frames are interpreted accordingly.

In the Offline mode the view can be zoomed and the frames can be browsed.

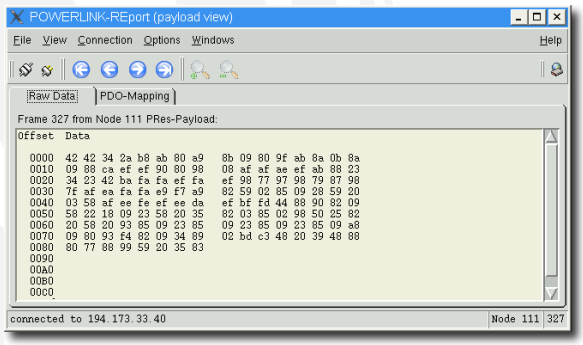
Further data like the source- or destination address and exact start and end times of the frames are displayed on request.

The number of displayed frames can be limited by special filters.

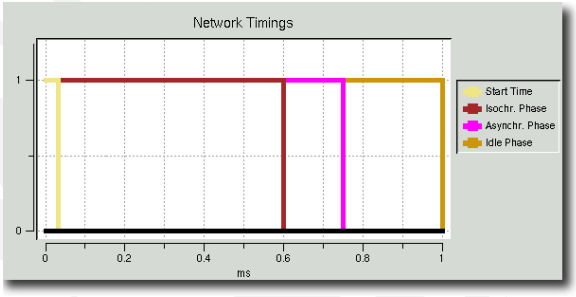
A list representation of all Ethernet frames of a cycle brings detailed information. Response times of the nodes and length of frames can be determined exactly, and NMT- and error information are displayed for each frame. All available information including the precise time stamp can be saved in the standardized OpenDocument format or as a CSV file for further processing.



The payload of POWERLINK nodes can be displayed as raw data or interpreted according to the PDO-Mapping. The raw data view displays the data in hexadecimal notation. The first row contains the offset from the start of the payload data.

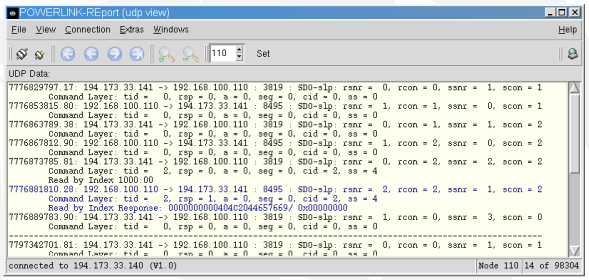


Besides the display of their chronological development, the characteristics can be evaluated statistically. The arithmetic mean, minima and maxima and the dispersion of the values be can recorded and evaluated by the POWERLINK-RE*port*.



Likewise the different phases of an POWERLINK cycle can be monitored.

Complete POWERLINK cycles, messages with payload or statistical information can be recorded in the memory or in a file. These data can be evaluated in detail afterwards.



A special feature of the POWERLINK-RE***port*** is the interpretation of SDO messages. The sequence layer and command layer protocol is analyzed in detail and all important information is extracted. Additionally, different SDO message types are displayed with different colors.

Diagnosis

With it numerous diagnostic features the POWERLINK-RE***port*** can be used for the diagnosis of running networks. A monitoring of network parameters allows the preventive detection of problems with the network and prevents production downtimes in advance. By diagnosis during production severe damages to machines and financial loss can be avoided.

POWERLINK-Interface

As hardware to access an POWERLINK network an POWERLINK Gateway is taken advantage of. Special requirements are made to the timely exact acquisition of Ethernet frames. The gateway has a second Ethernet TCP/IP port for the evaluation software, the POWERLINK-RE***port*** GUI.

System Requirements

Operating System:	Windows™ 2000, XP, Vista LINUX™
Processor:	Pentium IV or later
RAM:	256 MByte
Harddisk Space:	>25 MByte available

Ordering Information

1740/10	POWERLINK-REport-W (Full Version, Windows™)
1740/20	POWERLINK-REport-L (Full Version, LINUX™)

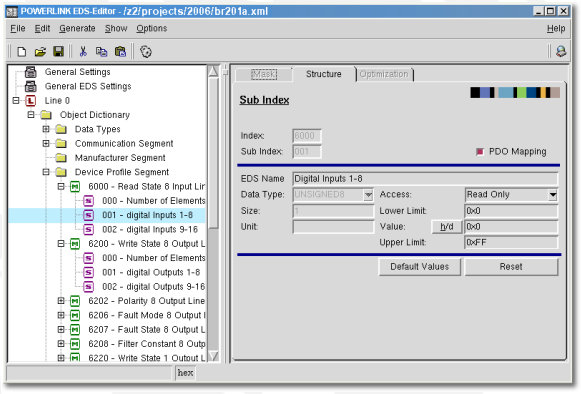
Functional demo versions of the software tools are available for download on our web site *http://www.epl-tools.com/shop.html*.

The POWERLINK-RE***port*** was awarded with the German prize

POWERLINK EDS-Editor

Overview

The POWERLINK *EDS-Editor* is a tool for the rapid and cost-saving creation of **Ethernet POWERLINK** device description files. Furthermore it creates a detailed documentation of the parameters in HTML.



Description

With the POWERLINK *EDS-Editor* a tool is available, which frees the developer from error-prone and repetitive tasks. It guarantees the consistency of the Electronic Data Sheet and the documentation. By default the POWERLINK communication profile is delivered with the tool. In addition to the POWERLINK specific communication database other databases are provided for the CANopen device profiles CiA 401, CiA 402, CiA 406 and CiA 417. The POWERLINK *EDS-Editor* eases the first step into the POWERLINK protocol and device development is accelerated.

EDS File

The EDS or XML Device Description(XDD) belongs to the standard scope of delivery of every POWERLINK device. It is an XML file according to ISO 15745 and it describes all relevant information about the application (POWERLINK device). These files are used by configuration tools and PLCs to integrate the devices in POWERLINK networks. The POWERLINK *EDS-Editor* supports the description of the following ISO 15745 entities:

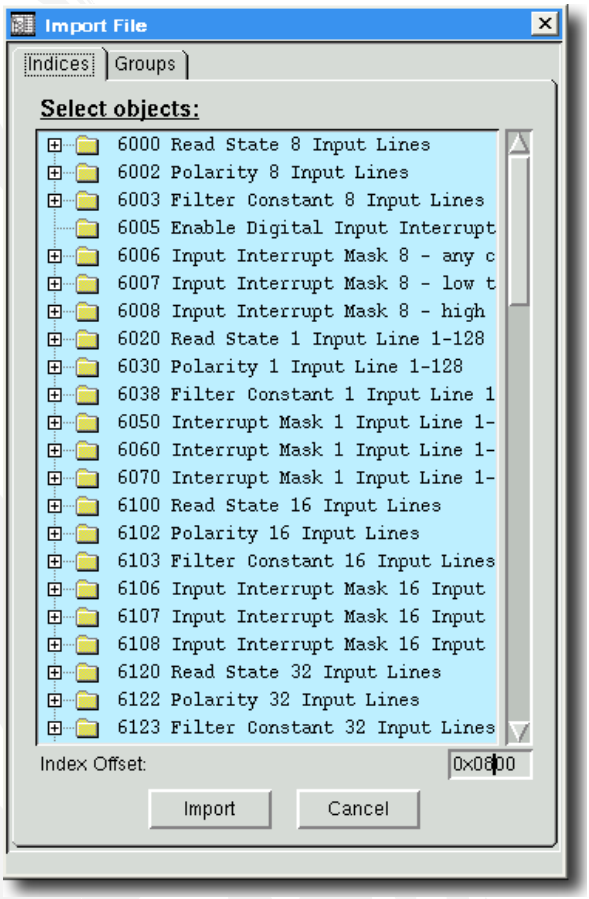
- DeviceIdentity,
- DeviceFunction,
- ApplicationLayers and
- NetworkManagement-Layer.

Documentation

On the generated object dictionary a documentation is created. Each device parameter is listed with its attributes and description. Output format is HTML and plain text. This documentation can serve as specification at the beginning of development and be brought into the user documentation, as well as into the marketing documents.

Profile databases

The POWERLINK *EDS-Editor* supports the import of profile databases containing templates for the objects and the import of EDS files. Therewith the POWERLINK *EDS-Editor* allows the re-use of existing know-how.


Import File
Indices
Groups
Select objects:
6000 Read State 8 Input Lines
6002 Polarity 8 Input Lines
6003 Filter Constant 8 Input Lines
6005 Enable Digital Input Interrupt
6006 Input Interrupt Mask 8 - any c
6007 Input Interrupt Mask 8 - low t
6008 Input Interrupt Mask 8 - high
6020 Read State 1 Input Line 1-128
6030 Polarity 1 Input Line 1-128
6038 Filter Constant 1 Input Line 1
6050 Interrupt Mask 1 Input Line 1-
6060 Interrupt Mask 1 Input Line 1-
6070 Interrupt Mask 1 Input Line 1-
6100 Read State 16 Input Lines
6102 Polarity 16 Input Lines
6103 Filter Constant 16 Input Lines
6106 Input Interrupt Mask 16 Input
6107 Input Interrupt Mask 16 Input
6108 Input Interrupt Mask 16 Input
6120 Read State 32 Input Lines
6122 Polarity 32 Input Lines
6123 Filter Constant 32 Input Lines
Index Offset:
0x0800
Import
Cancel

Import mask for a profile database

Generation

After an automatic check of the data for consistence, missing entries or incompatible data types, the following files are created:

- the XML EDS file,
- documentation as HTML and plain text,
- optionally an EDS file using the previous CANopen format.

If the check fails, errors and warnings are written into a log file.

Profile Databases

The profile databases contain the objects of the corresponding profile. Each object is configured according to the standard and is explained in detail. For every kind of objects in a profile a prototype is included and this object can be cloned by copying or duplicating. By importing these prefabricated objects into the own project the development time can be reduced and the risks of a manual implementation are minimized.
The following profile databases are currently available:

- PL - POWERLINK communication profile
- CiA-401 - Device Profile for Generic I/O Devices
- CiA-402 - Device Profile for Drives and Motion Control
- CiA-404 - Device Profile for Measuring Devices and Closed-Loop Controllers
- CiA-406 - Device Profile for Encoders
- CiA-417 - Application Profile for Lift Control Systems

Advantages

- acceleration of the development by avoidance of double work
- error prevention during creation of the object dictionary and EDS
- consistent documentation to the respective device version
- re-use of the data for similar devices
- supply of the data for standardized POWERLINK device profiles

System Environment

The POWERLINK *EDS-Editor* runs on PC's with

- Operating System: Windows™ 2000, XP, Vista and LINUX™
- Processor: Pentium or later
- RAM: > 256 MByte, HD > 20 MB available

Ordering Information

1761/10	POWERLINK-XML-ED-W (Full Version Windows™)
1761/20	POWERLINK-XML-ED-L (Full Version LINUX™)

Functional demo versions of the software tools are available for download on our web site *http://www.epl-tools.com/shop.html*.

POWERLINK Managing Node for PCI Bus

Overview

The POWERLINK Managing Node for PCI Bus is a PC interface card, which connects a PC with an Ethernet Powerlink network. The complete Managing Node functionality according to POWERLINK standard V2.0 is integrated on the device. With the device you can send data from the PC into the POWERLINK network or log the POWERLINK traffic on the PC. Thereto *port* offers a special software interface for the PC-based configuration tool POWERLINK Device Monitor, as well as an interface for the analysis tool POWERLINK-REport.

Description

The card, which is to be plugged into a free PCI Slot of the PC, consists of three Ethernet ports. You can configure up to two real-time or generic ports. The third port is linked with an intern hub to one of the two networks and belongs consequently to this network.

The three Ethernet connections are labeled 0, 1 and 2. Connection 0 has its own controller called "Interface 0" while connections 1 and 2 are connected to the second ethernet controller with an integrated hub called "Interface 1".

All communication between the host system and the local processor is done via the Dual-Port-RAM (DPR), which has a size of 512KB. Every Ethernet port has two LEDs, which indicates a link resp. network activity. Furthermore the card has several status and error LEDs. With an address switch the user can set the node id of the device. So the card can act as a Controlled Node too. Every node id unequal to 240 causes the function as Controlled Node.

The object indices are references to the **Ethernet POWERLINK** specification. Most objects of the Managing Node (MN) are mapped directly into the Dual-Ported-Ram. The Managing Node supplies buffers for a direct exchange of objects with the Controlled Nodes (CN) and a large buffer for asynchronous TCP/IP communication with CNs. The MN may act as an POWERLINK-Router appearing to the host as a network adapter.

Technical Data

Supply voltage	5V; 3.3V
Power consumption	Max. 12W
Bus Frequency	33MHz
Dimensions	100x170x21 mm (HxLxD)
Mounting	PCI Slot
Weight	approx. 150g
Temperature range	0°C - 50°C (opt. -20°C - 70°C)

Ordering Information

0552/01 POWERLINK-PCI-SHF

POWERLINK Gateway

Overview

port's universal POWERLINK Gateway is a small box with two Ethernet ports. The real-time port is connected to an POWERLINK network. The other port acts as an interface to an IP network. Running SND's HyNetOS operating system and POWERLINK V2.0 stack it can be used either as CN-based gateway/router or as MN-based gateway/router. With this Gateway an POWERLINK network can be integrated seamlessly into an intranet, allowing to access POWERLINK devices from every PC in your office. Any PC connected can thus reach any Powerlink node via TCP/IP (ping, HTTP, FTP etc.) without disturbing the real-time cycle.

Optionally *port* offers special software interfaces for PC-based configuration tools, like POWERLINK Device Monitor, as well as an interface for the analysis tool POWERLINK-REport.

Description

The gateway consists of two units with one Micro WebTarget MWT2/XS Board each. Every board has an 128MHz E1-16XS processor from Hyperstone and an Ethernet-Controller LAN91C111 from SMSC installed. The two units communicate over a Dual Ported RAM (DPR). Here the IDT70V05S/L from IDT is used. The compact casing with its dimensions of 23x100x115 mm is mounted as top-hat rail with integrated grounding.

Every Ethernet port has two LEDs, which indicates a link resp. network activity. Furthermore the POWERLINK side has a status (BS) and an error (BE) LED, which can be controlled by software. With an address switch (ID) the user can set the node id of the device. Thus he can switch the node type easily. Is the node id equal to 240 the device works as managing node. Every other id causes the function as controlled node.

The ip address of the non POWERLINK side can be set using a TCP based tool, which is similar to telnet. With a switch you can shift between configured and standard IP address (10.10.10.10). On the POWERLINK side there is also such a switch to shift between the IP addresses 192.168.100.[node id] and 10.10.10.10.

Technical Data

Temperature range	0°C - 55°C
Supply voltage	24 VDC
Dimension	23x100x115 mm (BxHxD)

Ordering Information

0551/01 POWERLINK-DTGW

POWERLINK PE²MAC

Overview

The PE²MAC developed by ***port*** is a special MAC controller optimized for Ethernet Powerlink for Xilinx-FPGAs.

Through several special transmit buffers and the possibility to respond automatically to POWERLINK frames by hardware, extremely short response times are reached. Special receive filters allow pre-filtering of POWERLINK messages by the MAC controller and releases the software from this tasks. These filters support the pre-selection of Ethernet frames, that are needed by the processing software. Based on the used settings only those frames that match the filters are stored in the receive buffers. This mechanism releases the CPU from processing frames not relevant for the software.

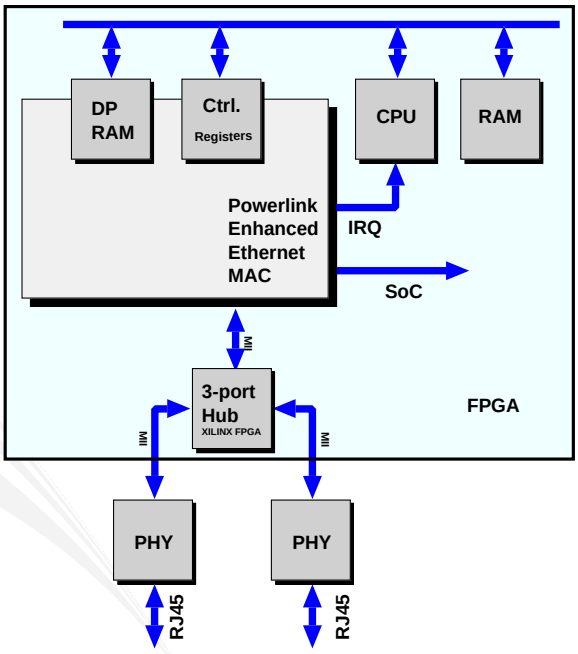
For the MAC unit a VHDL firmware design was developed that is optimized for POWERLINK data transfer.

Description

With FPGAs custom-made, scalable and future-proof solutions open up for the equipment manufacturer. The reusability of IPs, quick time to market and high cost efficiency, are substantial advantages of a FPGA based solution. E.g. a solution with Industrial Ethernet interface, hub logic and processor core can be implemented in one single XILINX-FPGA. Furthermore the embedded processor can directly be implemented in the FPGA.

The PE²MAC from ***port*** works in a Xilinx-FPGA together with a Microblaze processor. The processor interface is 32-bit wide.
The PE²MAC is a standalone component that is linked to the processor via registers (control- and status information), a dual-port-memory (data buffer) and an interrupt request signal. Furthermore a separated signal is provided that signals an incoming SoC packet.

A special PE²MAC driver provides an optimal interconnection to ***port***'s POWERLINK *Library*.



In connection with the POWERLINK-HUB not only a straightforward connection to an POWERLINK network is provided but also a sophisticated solution with technical parameters (short response times, fast reaction to network events) that conforms to the requirements of the POWERLINK standard is available.

Technical Data

number of required blockRAMs	6
platform	Xilinx-FPGA
processor-connection	On-chip Peripheral Bus (OPB)
curcuit model	OPB-Slave
operation mode	100MBit
PHY interface	MII

Ordering Information

1712/00 PE²MAC-VHDL

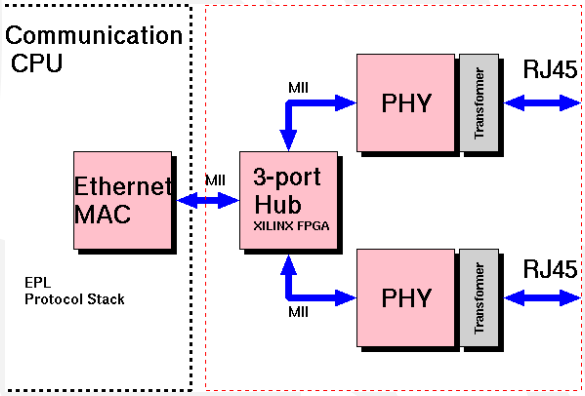
POWERLINK Hub VHDL Code

Overview

The POWERLINK hub from ***port*** is a generic hub design for FPGA and for CPLDs written in VHDL. The propagation delay is less than 390ns. In conjunction with ***port***'s Enhanced Ethernet MAC it offers a fast and easy connection to an POWERLINK network.

Description

The POWERLINK hub from ***port*** is a generic hub design for for FPGA and CPLDs written in VHDL. It is compliant to the IEEE 802.3u standard and supports 100MBit half duplex connections. The propagation delay below 390ns is far inferior to the time, which is claimed in the IEEE 802.3 standard.



The 3-port hub features two external and one internal interface. The number of PHY ports can be expanded easily. By integrating the hub in an POWERLINK device, it can be simply integrated in POWERLINK networks with a daisy chain. An external hub is not required and extensive cabling is not applicable. In conjunction with ***port***'s POWERLINK MAC, a highly optimized VHDL POWERLINK MAC controller for Xilinx-FPGAs, a fast connection to your POWERLINK network is reached. This results in fast response times of Controlled Nodes.

Technical Data

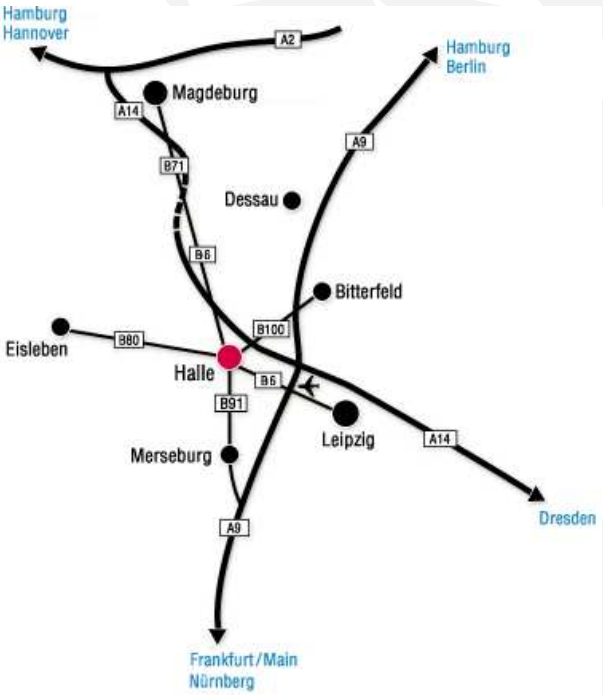
Operating mode	100Mbit/s half duplex
Number of ports	2 external, 1 internal (expandable)
propagation delay	<=390ns
prop. delay without PHYs	<=120ns
Number of flip-flops	150
Number of slices	158
Jitter on total HUB delay	40ns
maximum difference of time domain frequencies	166ppm
PHY interface	MII

Ordering Information

1711/00	POWERLINK-Hub(I)-VHDL
1711/01	POWERLINK-Hub(S)-VHDL

Directions to port GmbH

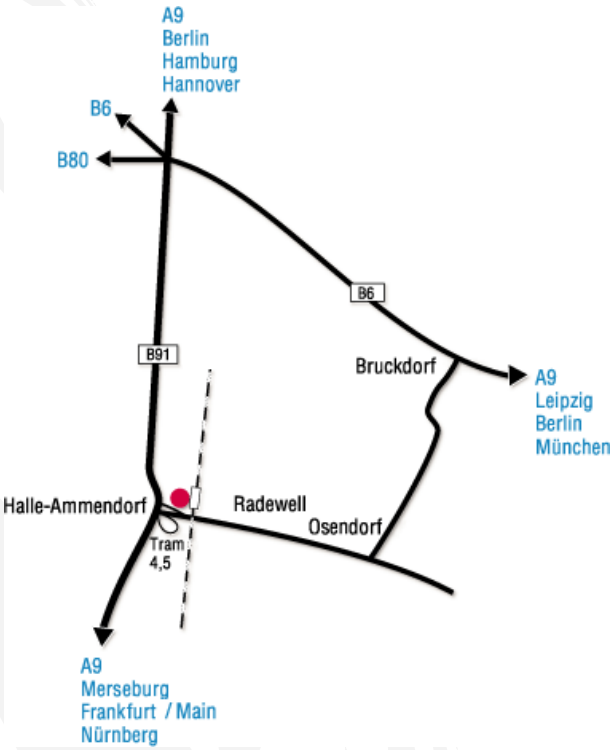
We recommend for planning your journey to ***port*** to use a route planner for example Map24 ⟨http://www.port.de/engl/contact/journey.html⟩. Or you can pursue the plan on our web site for your journey to Halle/Saale in the state Saxony-Anhalt. The next airport for your journey by air is the airport Halle/Leipzig.



Our business premises are located in the southern part of Halle. This district of Halle/Saale is called Ammendorf. The easiest way is from the south on the main road B91. In addition, coming from other directions you arrive at Ammendorf directly by all means on the main road B91 in a southward direction.

Overnight Stay

We recommend our guests the Hotel Eigen ⟨http://www.hotel-eigen.de⟩, approximately 10 min walking distance from ***port.*** You find further overnight accommodations under the City Halle web site ⟨http://www.halle.de⟩.



On the city map of Ammendorf you can see how to find ***port***-GmbH there.

Parking Places

Parking places are available on the road in front of the business premises or on the parking place at the back-side of the house.